University of Exeter
College of Engineering, Mathematics and Physical Sciences

The Analysis and Application of Artificial Neural Networks for Early Warning Systems in Hydrology and the Environment

Andrew Paul Duncan

September 2014

Supervised by Dr. Edward C. Keedwell and Prof. Dragan Savić

Submitted by Andrew Paul Duncan to the University of Exeter as a thesis for the degree of Doctor of Philosophy in Computer Science in September 2014

Signature: ……..

Acknowledgment

I would like to thank my supervisors Dr. Edward Keedwell and Professor Dragan Savić for their continued patience and the support and guidance that they have given me throughout my PhD, for which I will always be very grateful. I would also like to thank Dr Chris Ferro for his guidance and mentoring, which are much appreciated. Thank you also to Professor Dawei Han, Professor Richard Everson and Dr Yiming Ying for kindly agreeing to be my examiners. Thank you to my work supervisor, Professor Fayyaz Ali Memon for patience and forbearance.

Thanks especially to my sister, Jane Duncan, for her continued support, advice and confidence in my ability to complete; to my friends Dr Michael Halls, Elinor Scott and Stig Rune Framsteg, for your support, wisdom, kindness and forbearance over this time. Special thanks to my three colleagues in our office at university: Dr Albert Chen, Dr Michael Hammond and Dr Kourosh Behzadian for continued advice and support; also to so many colleagues, friends and family too numerous to mention who have given continuous encouragement over the whole period of the PhD – all of which is very much appreciated.

Thanks to Lars Werntz for inspiring me to return to academia – encouragement that I will always remember.

Last but not least, thanks to my dog Zack for being a continuous and very patient companion during the writing up of this thesis. Walkies!

Abstract

Artificial Neural Networks (ANNs) have been comprehensively researched, both from a computer scientific perspective and with regard to their use for predictive modelling in a wide variety of applications including hydrology and the environment. Yet their adoption for live, real-time systems remains on the whole sporadic and experimental. A plausible hypothesis is that this may be at least in part due to their treatment heretofore as “black boxes” that implicitly contain something that is unknown, or even unknowable. It is understandable that many of those responsible for delivering Early Warning Systems (EWS) might not wish to take the risk of implementing solutions perceived as containing unknown elements, despite the computational advantages that ANNs offer.

This thesis therefore builds on existing efforts to open the box and develop tools and techniques that visualise, analyse and use ANN weights and biases especially from the viewpoint of neural pathways from inputs to outputs of feedforward networks. In so doing, it aims to demonstrate novel approaches to self-improving predictive model construction for both regression and classification problems. This includes Neural Pathway Strength Feature Selection (NPSFS), which uses ensembles of ANNs trained on differing subsets of data and analysis of the learnt weights to infer degrees of relevance of the input features and so build simplified models with reduced input feature sets.

Case studies are carried out for prediction of flooding at multiple nodes in urban drainage networks located in three urban catchments in the UK, which demonstrate rapid, accurate prediction of flooding both for regression and classification. Predictive skill is shown to reduce beyond the time of concentration of each sewer node, when actual rainfall is used as input to the models.

Further case studies model and predict statutory bacteria count exceedances for bathing water quality compliance at 5 beaches in Southwest England. An illustrative case study using a forest fires dataset from the UCI machine learning repository is also included. Results from these model ensembles generally exhibit improved performance, when compared with single ANN models. Also ensembles with reduced input feature sets, using NPSFS,

demonstrate as good or improved performance when compared with the full feature set models.

Conclusions are drawn about a new set of tools and techniques, including NPSFS and visualisation techniques for inspection of ANN weights, the adoption of which it is hoped may lead to improved confidence in the use of ANN for live real-time EWS applications.

Table of Contents

List of Tables

List of Figures

List of publications

Duncan, A., Chen, A.S., Keedwell, E., Djordjevic, S., Savic, D.A., 2011. Urban flood prediction in real-time from weather radar and rainfall data using artificial neural networks, in: IAHS Red Book series no. 351, 58. Presented at the Weather Radar and Hydrology International Symposium, International Association of Hydrological Sciences, Exeter, UK.

Duncan, A.P., Chen, A.S., Keedwell, E.C., Djordjevic, S., Savic, D.A., 2013. RAPIDS: Early Warning System for Urban Flooding and Water Quality Hazards (Extended Abstract), in: AISB 2013. Presented at the Artificial Intelligence and Simulation of Behaviour Conference; Machine Learning in Water Systems Symposium, AISB, University of Exeter, pp. 25–29.

Duncan, A.P., Keedwell, E.C., Djordjevic, S., Savic, D.A., 2013a. Early Warning System for Bathing Water Quality (Poster), in: Bathing Waters 2013. Presented at the Bathing Waters 2013, Defra, England, Southport, UK.

Duncan, A.P., Keedwell, E.C., Djordjevic, S., Savic, D.A., 2013b. Machine Learning-Based Early Warning System for Urban Flood Management, in: International Conference on Flood Resilience: Experiences in Asia and Europe. Presented at the International Conference on Flood Resilience 2013, University of Exeter, Exeter, UK, pp. 237–238.

Duncan, A.P., Tyrrell, D., Smart, N., Keedwell, E.C., Djordjevic, S., Savic, D.A., 2013. Comparison of machine learning classifier models for bathing water quality exceedances in UK, in: IAHR35. Presented at the IAHR35, IAHR, Chengdu, China, p. In press.

Guidolin, M., Duncan, A., Ghimire, B., Gibson, M., Keedwell, E.C., Djordjevic, S., Savic, D., 2012. CADDIES: a new framework for rapid development of parallel cellular automata algorithms for flood simulation, in: Hydroinformatics International Conference. Presented at the HIC 2012, IWA, Hamburg, Germany.

Guidolin, M., Duncan, A., Keedwell, E.C., Chen, A.S., Djordjevic, S., Savic, D.A., 2011. Design of a graphical framework for simple prototyping of pluvial flooding cellular automata algorithms, in: Proceedings of CCWI 2011. Presented at the Computing and Control for the Water Industry 2011. “Urban Water Management - Challenges and Opportunities”, University of Exeter.

Savić, D.A., Bicik, J., Morley, M.S., Duncan, A., Kapelan, Z., Djordjevic, S., Keedwell, E.C., 2013. Intelligent Urban Water Infrastructure Management. JIISc, Water Management in Changing Environment 93, 319–335.

Schellart, A., Ochoa, S., Simões, N., Wang, L.P., Rico-Ramirez, M., Liguori, S., Duncan, A., Chen, A.S., Keedwell, E., Djordjević, S., others, 2011. Urban pluvial flood modelling with real time rainfall information–UK case studies, in: ICUD 2011. Presented at the 12nd International Conference on Urban Drainage, IWA, Porto Alegre/Brazil.

List of Acronyms

	Acronym	Novel?	Definition / Description
1-9			
	1HL		1 Hidden Layer neural network
	2HL		2 Hidden Layers neural network
A			
	ACF		Auto Correlation Function
	ACO		Ant Colony Optimisation
	ANaNAS	Yes	Artificial Neural Network Analysis System (MS Excel workbook-based)
	ANFIS		Adaptive Neuro-Fuzzy Inference System
	ANN		Artificial Neural Network
	AR		Auto Regressive linear model
	ARMA		Auto-Regressive Moving Average linear model
	Artmap		Adaptive Resonance Theory Mapper
	AuC		Area under the Curve (of an ROC plot)
B			
	B_{A1}	Yes	Accuracy Band for peak flood depth categories (below manhole cover)
	B_{A2}	Yes	Accuracy Band for peak flood depth categories (above manhole cover)
	BBN		Bayesian Belief Network
C			
	CART		Classification and Regression Trees method
	CCF		Cross Correlation Function
	CELS		Cooperative Ensemble Learning System
	CFS		Correlation-based Feature Selection
	CNPSA	Yes	Combined Neural Pathway Strength Analysis
	CO-TREC		Continuity of Tracking Radar Echoes by Correlation
	COWAMA		Coastal Water Management project (Spain)
	CSO		Combined Sewer Overflow safety device
	CWS		University of Exeter Centre for Water Systems
D			

	Acronym	Novel?	Definition / Description
	DC		Drought Code index from the FWI system
	DDM		Data Driven Model
	Defra		Department for Environment Food and Rural Affairs (UK government)
	DM		Decision Maker (human)
	DMC		Duff Moisture Code index from the FWI system
	DO		Dissolved Oxygen
	DSS		Decision Support System
	DT		Decision Tree model
	DWF		Dry Weather Failure
E			
	EA		Evolutionary Algorithm
	EA-E		Environment Agency for England
	E_{AP}	Yes	Amplitude error of hydrograph peak
	E-coli		Escherichia coli - potentially pathogenic bacterium species
	ED		Euclidean Distance
	EMD		Empirical Mode Decomposition
	E_{OPT}	Yes	Optimum Euclidean distance
	EP		Evolutionary Programming
	EPA		Environment Protection Agency (USA or Scotland)
	EPS		Ensemble Prediction Systems
	EQR	Yes	Ensemble interQuartile Range
	ES		Evolutionary Strategies
	ETAO		Ensemble Transformation and Adaptive Observations
	E_{TP}		Timing error of hydrograph peak
	E_{TV}		Total Volume Error
	EWS		Early Warning System
F			
	FC		Faecal Coliforms - potentially pathogenic bacterium species
	FFBP		Feed Forward Back Propagation - optimisation

	Acronym	Novel?	Definition / Description
			algorithm for ANNs
	FFMC		Fine Fuel Moisture Code (from FWI index)
	FiS		Fisher Score
	FL		FLooding (for sewer manholes)
	FM		F-Measure - measure of goodness of a classifier model
	FN		False Negatives
	FNR		False Negative Ratio
	FP		False Positives
	FPR		False Positive Ratio
	FRFS		Fuzzy-Rough Feature Selection
	FRMRC		Flood Risk Management Research Consortium
	FS		Faecal Streptococci - potentially pathogenic bacterium species
	FWI		Canadian Forest Fire Weather Index
G			
	GA		Genetic Algorithm
	GD		Gradient Descent - approach to optimisation of ANNs
	Genitor		GENetic ImplemenTOR, a genetic search algorithm
	GEP		Gene Expression Programming
	GIS		Geographical Information System
	GLUE		Generalised Likelihood Uncertainty Estimation methodology
	GP		Genetic Programming
H			
	HATS		Harbour Area Treatment Scheme (Hong Kong)
	HS		Harmony Search algorithm
	HSFS		Harmony Search Feature Selection
	HydroMAT	Yes	Hydrographical Model Analysis Tool
	HypE		Hypervolume-based many-objective Evolutionary algorithm
I			

	Acronym	Novel?	Definition / Description
	ICA		Independent Component Analysis
	ICFR		International Conference on Flood Resilience
	IE		Intestinal Enterococci - potentially pathogenic bacterium species
	ISI		Initial Spread Index from the FWI system
J			
K			
	KNN		Kohonen Neural Network (aka Self Organising Map)
	K-NN		K - Nearest Neighbours method of classification
L			
	LCS		Learning Classifier System
	LOOCV		Leave One Out Cross Validation
M			
	MATLAB		MATrix LABoratory - programming language
	M_{C1}	Yes	Confusion Matrix for peak flood depth categories (below manhole cover)
	M_{C2}	Yes	Confusion Matrix for peak flood depth categories (above manhole cover)
	MCMC		Monte-Carlo Markov-Chain simulation approach
	MF		Manhole Flooding
	MI		Mutual Information
	MLH		Maximum LikeliHood
	MLP		Multi-Layer Perceptron (ANN)
	MLR		Multiple Linear Regression model
	MOEA		Multi Objective Evolutionary Algorithm
	MOI		Maximum Output Information
	MOO		Multi Objective Optimisation
	MS		Manhole Surcharging
	MSE		Mean Squared Error
	MSEREG		Mean Squared Error with Weight Regularisation term applied
	MutI		Mutual Information

	Acronym	Novel?	Definition / Description
N			
	NAPI		New Antecedent Precipitation Index
	NCL		Negative Correlation Learning
	NDVI		Normalized Difference Vegetation Index
	NE		Neuro-Evolution (use of EAs to train ANNs)
	NFCV		N-Fold Cross Validation
	NFIR		Non-linear Finite Impulse Response
	N_{HU}	Yes	Number of Hidden Units
	NID		Network Interpretation Diagram
	N_{IN}	Yes	Number of ANN INputs
	NN		Neural Network
	NPSD	Yes	Neural Pathway Strength Diagram
	NPSFS	Yes	Neural Pathway Strength Feature Selection
	NRMSD		Normalised Root Mean Squared Deviation
	NS		Nash Sutcliffe efficiency coefficient
	NSEC		Nash Sutcliffe Efficiency Coefficient
	NSGA-II		Non-dominated Sorting Genetic Algorithm II
	NWD	Yes	Nguyen-Widrow ANN initialisation (all ensemble members Different)
	NWP		Numerical Weather Prediction
	NWS	Yes	Nguyen-Widrow ANN initialisation (all ensemble members Same)
O			
	OBS		Optimal Brain Surgeon
	OF		Objective Function
	OJR	Yes	Olden and Jackson Randomisation
	OS		Ordnance Survey (UK)
P			
	PACF		Partial AutoCorrelation Function
	PBIAS		Percentage BIAS
	PCA		Principal Component Analysis - dimension reduction technique
	PCFS		Probabilistic Consistency-based Feature Selection

	Acronym	Novel?	Definition / Description
	PI		Partial Information
	PMI		Partial Mutual Information
	PSL		Parametric Simple Linear model
	PSO		Particle Swarm Optimisation
	PTA	Yes	Prediction Timing Advance
Q			
	QPF		Quantitative Precipitation Forecasting
R			
	RAPIDS	Yes	RAdar Pluvial flooding Identification for Drainage System
	RBF		Radial Basis Function
	rBWD		revised Bathing Water Directive (European Commission)
	Reduct		Reduced inpuT feature set
	RELIEF / RELIEF-F		The Relief feature selection algorithm and one of its successors
	RH		Relative Humidity
	RMSE		Root Mean Squared Error
	RNN		Recursive Neural Network
	ROC		Receiver Operating Characteristic curve (chart)
	RP		Return Period - for rainfall events
	RTM		Real Time flood Management (UKWIR project)
S			
	SCG		Scaled Conjugate Gradients
	SFRB		Sustainable Flood Retention Basin
	SOEA		Single Objective Evolutionary Algorithm
	SOM		Self Organising Map (aka Kohonen Neural Network)
	STCA		Short Term Conflict Alert (air traffic control)
	STW		Sewage Treatment Works
	SU		Surcharge (relating to manhole flooding)
	SVM		Support Vector Machine
	SWAT		Soil and Water Assessment Tool

	Acronym	Novel?	Definition / Description
	SWMM		Storm Water Management Model
T			
	TAerr	Yes	Time-Amplitude Error measure
	TN		True Negatives
	TNR		True Negative Rate
	ToC		Time of Concentration (of a catchment or sewer network)
	TP		True Positives
	TPR		True Positive Rate
	TREC		Tracking Radar Echoes by Correlation
U			
	UCI		University of California at Irvine
	UDD	Yes	Uniform Distributed ANN weight initialisation (all ensemble members Different)
	UDS	Yes	Uniform Distributed ANN weight initialisation (all ensemble members Same)
	UKWIR		United Kingdom Water Industry Research organisation
	URN		Unique Reference Number (for beach identification)
V			
	VB		Virtual Beach - water quality prediction system
W			
	WDN		Water Distribution Network
	WHO		World Health Organisation (part of the United Nations)
	Wio	Yes	Matrix of ANN Weights (from Input to Output)
	WWTP		Waste Water Treatment Plant
	WWTW		Waste Water Treatment Works
X			
	Xcorr		Cross Correlation
Y			
Z			

Chapter 1: Introduction

1.1 Motivation and aims

Despite comprehensive research into Artificial Neural Networks (ANNs) both from a computer scientific perspective and for their application to predictive modelling in hydrology and the environment, their adoption for live, real-time systems remains on the whole sporadic and experimental. This may be at least in part due to their treatment heretofore as “black boxes” that implicitly contain something that is unknown, or even unknowable. It is understandable that many of those responsible for delivering Early Warning Systems (EWS) might not wish to take the risk of implementing solutions perceived as containing unknown elements, despite the significant computational advantages that ANNs offer: rapid execution and the automation of model calibration amongst these.

The overall aim is to make a contribution to understanding of ANNs as predictive data-driven modelling (DDM) tools in the hope of increasing their application to much needed live real-time early warning systems, for example for flood or bathing water quality risks as well as a wider set of predictive modelling scenarios.

This thesis therefore aims to build on existing research that opens up the box:

- To develop tools and techniques that analyse and use ANN weights and biases especially from the viewpoint of neural pathways from inputs to outputs of feedforward networks
- To demonstrate novel approaches to self-improving predictive model construction for both regression and classification problems
- To provide a new computationally efficient algorithm for input feature selection employing ensembles of Artificial Neural Networks (ANNs) that can automatically select a subset of relevant inputs from an entire input feature set through the analysis of the learned network weights
- To establish that such an algorithm is applicable to a potentially wide set of machine learning modelling problems, by demonstrating its use for a difficult prediction problem from the standard UCI repository

- To introduce and demonstrate neural pathway-strength visualisation techniques, which assist with opening up the black box and reveal structure in the learnt ANN weights
- To explore the use of multi-output ANNs for predictive modelling of multiple quantities simultaneously. This is applied to prediction of flooding at multiple nodes in urban drainage networks
- To conduct a sensitivity analysis to establish the limits of predictability for DDMs (such as ANNs) used with time-lagged inputs in an urban flooding prediction context.
- To develop ANN model ensembles that aim to improve on existing decision-tree (DT) and simple threshold models for the classification of bathing water quality, through the use of the receiver-operating characteristic (ROC) as a performance metric for ANN training
- To investigate two implementations of this:
    1. Using neuro-evolution in a dual-objective optimisation using NSGA-II
    2. Using Scaled Conjugate Gradients (SCG) algorithm (Møller, 1993; Press, 2007) in a single-objective area-under-the -ROC–curve (AuC) optimisation
- To evaluate the aggregate performance of such ensembles of models using majority voting and ensemble mean and median predictions.

1.2 Novelties

Within the limits of the author's understanding, the following aspects of this thesis are believed to be novel:

- An approach to visualisation of the overall net effect of each input on each output of an ANN; designated here as "Combined Neural Pathway Strength Analysis" (CNPSA)
- The creation of an ensemble of ANN models using N-fold cross-validation (NFCV) based on division of datasets into a number of folds and the development of a novel metric for measuring relevance of input features based on variability of CNPSA neural pathway strengths across all the members of the ensemble; this is labelled Ensemble interQuartile Range (EQR)

- The use of EQR created as above to determine the relevance of each input feature and permit automated feature selection by a meta-modelling process. This is referred to here as “Neural Pathway Strength Feature Selection” (NPSFS)
- A visualisation technique for viewing the internal operation of 2-layer feedforward neural networks during and following training. This reveals the structure of “morphemes” and “sememes” (Hinton, 1984; Hinton et al., 1993) within a 2-dimensional neural pathway strength space and its breakdown into three 2-dimensional subspaces organised by output neuron, hidden neuron or input signal. These are labelled "Neural Pathway Strength Diagrams" (NPSD)
- There are a number of potential benefits to this, including:
    - Mechanisms for pruning irrelevant connections
    - Improvement of model performance through such pruning
    - Increasing confidence in neural network models by non-expert practitioners, through providing tools for checking the ways models have made use of the information contained in the training dataset
    - An additional mechanism for evaluation of the relative effectiveness of ANN training algorithms.
    - Backtracing and faultfinding to identify root causes of problems with individual ANN models
- The use of multi-output ANNs to model urban flooding at multiple sewer nodes / locations simultaneously
- The use of ROC scenarios for the optimisation of ANNs for bathing water quality classifiers.

1.3 Structure of the thesis

The remaining chapters of this thesis are structured as follows:

Chapter 2 contains a literature review, where ANN research is reviewed and set into context within machine learning and optimisation. Neuroevolution – evolving ANNs using evolutionary algorithms – is also covered. Feature selection approaches are discussed as are ensemble modelling techniques. Approaches to visualisation of ANNs – especially with regard to their weights,

biases and/or neural pathways – are reviewed. Finally, applications of ANNs in the areas of urban flooding and bathing water quality prediction are reviewed and discussed.

Chapter 3 is a case study featuring the author's early work on the application of ANNs to urban flood prediction. Multi-output ANNs are used to model multiple nodes in urban drainage networks. Three case study urban catchments in southern England are used. A sensitivity analysis on the limits of prediction using ANNs with actual rainfall is included. A novel method of analysing ANN weights and biases is discussed and demonstrated.

Chapter 4 presents novel machine learning methodologies for input feature selection using ANN model ensembles. The method works via analysis of the weights learnt during training. Additionally, novel visualisation tools for ANNs are presented. The techniques are demonstrated using a dataset from the standard UCI machine learning repository.

Chapter 5 is a further case study in which the techniques developed in chapter 4 are applied to the problem of ANN classification of bathing water quality (in terms of exceedances of bacteria counts) at 5 beaches in south west England. These are also compared with simple threshold and decision tree models developed at the Environment Agency. Neuro-evolved and conventionally trained ANN ensembles using ROCs as performance functions are developed and evaluated.

Chapter 6 contains the conclusions of the thesis together with suggestions for further work.

Appendix A details the profiles and maps for each of the bathing beaches used in the chapter 5 case study, together with photos of the beaches taken by the author.

Chapter 2: Literature Review

2.1 Introduction

This chapter reviews existing research into techniques relevant to the novel methodologies described in this thesis. It is organised as follows:

Section 2.2 reviews Artificial Neural Networks (ANNs) and their applications. Section 2.3 concerns Evolutionary Algorithms (EAs) and applications. In section 2.4, EAs as a method for training ANNs are reviewed. Cross-validation techniques are covered in section 2.5, whereas section 2.6 explores feature selection approaches. In section 2.7, ensemble creation techniques are reviewed. Section 2.8 explores techniques for opening up the black box of the ANN and approaches to visualisation of their structure. Finally, section 2.9 covers prior applications of ANNs to urban flooding and bathing water quality, the two major subjects of case study included in chapters 4 and 5 of this thesis.

2.2 Artificial Neural Networks

2.2.1 Background

Artificial Neural Networks (ANNs) are networks consisting of a number of essentially identical neurons connected together using one of a number of possible architectures. Figure 2.3 illustrates a commonly-used ANN architecture. A huge body of research exists on ANNs and this machine learning approach is established; therefore a full description of the fundamentals of ANNs is not necessary here. However, a brief description is provided.

Historically, the concept of ANNs arose from early research into neurophysiology (Brodmann, 1909; Hebb, 1949; Hubel and Wiesel, 1963, 1961, 1959), which revealed that brains consist of specialised cells called neurons connected together in a network. Each neuron operates electrically and has a number of inputs (dendrites), a body (soma) and an output (axon), which is in contact with the dendrites of other neurons. These connections are referred to as synapses. The synaptic connectivity is regulated by chemical

neurotransmitters that determine strengths and senses (excitatory / inhibitory) of the connections. The neuron is observed to fire[1] when the weighted sum of its inputs exceeds a threshold value. The synaptic connection weightings are observed to be positively reinforced by frequent stimulation by incoming electrical pulses from the connected axons of other neurons (Bruner, 1957; Hebb, 1955; Rochester et al., 1956). Figure 2.1 shows a schematic diagram of a biological neuron.

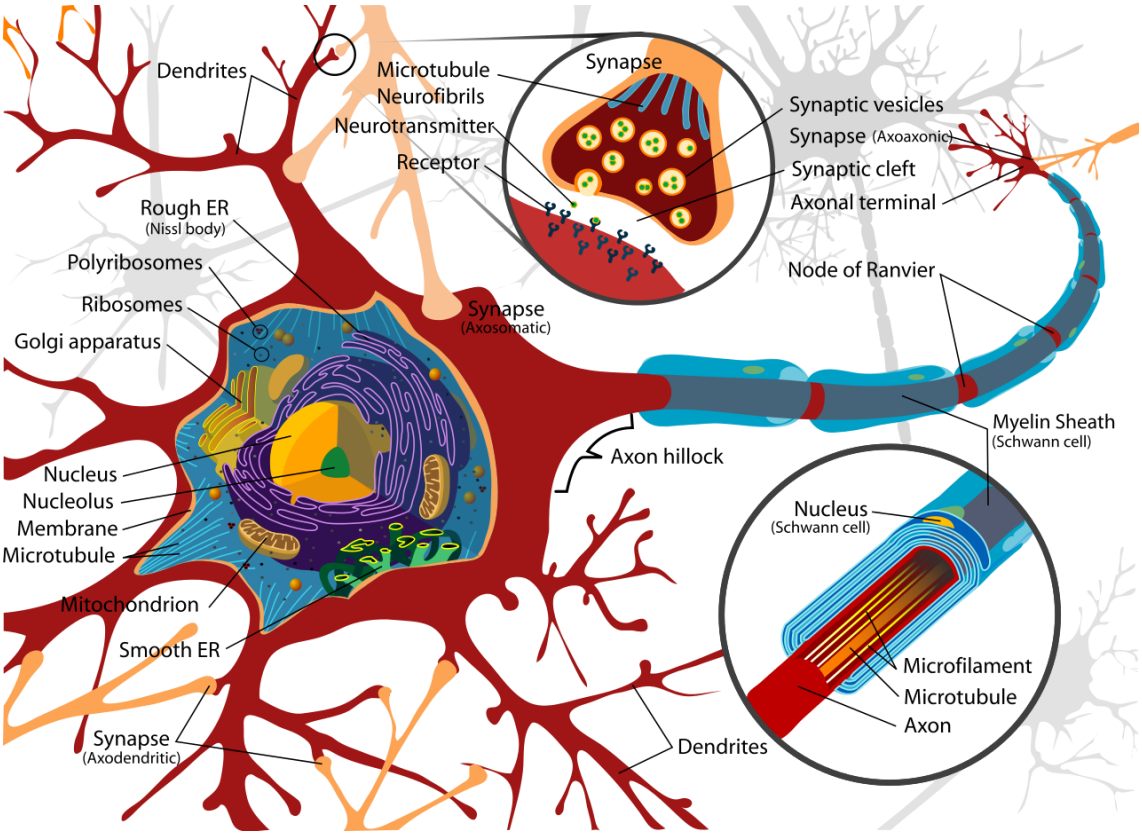


Figure 2.1. Schematic of biological neuron (Wikimedia, 2015)

The question of the possibility of construction of artificial networks to emulate the operation and learning capabilities of brains thus arose. Early researchers (Harmon, 1961, 1959; Hiltz, 1963; Rosenblatt, 1958) generally implemented ANNs in hardware due to lack of availability of many general purpose computers, whereas today the majority of ANNs are implemented in software. Despite this, early software simulations were also carried out (Rosenblatt, 1960). Regardless, each artificial neuron consists of a set of inputs with multiplicative weights and a single bias, a summation function, followed by an output activation function. It therefore performs the computation:

[1] *Generate electrical spike signals that travel along the axon*

$$y = f(x) = \kappa\left(\sum_{i}^{N} w_i\, x_i + \; b\right) \qquad (1)$$

where: x_i is the *i-th* input to the neuron, w_i is a weight associated with input *i*, *b* is a time-invariant bias level and *κ* is an activation function applied to the output of the neuron. This might typically implement the hyperbolic tangent (*tanh*), logistic sigmoid ($1/(1+e^{-x})$), a threshold switch or a linear function. The activation function is selected based on the type of data being processed and network being created, so selection is problem-specific. A threshold switch would output an all-or-nothing response whereas the linear and hyperbolic tangent functions would output floating point values. For a full treatment of these, the reader is referred to section 4 onwards in Sontag (1993). Figure 2.2 provides a conceptual model of the operation of an artificial neuron that implements the function corresponding to equation (1).

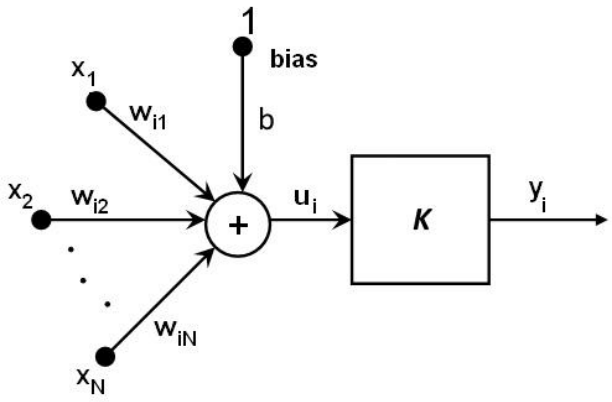


Figure 2.2. Schematic of artificial neuron

The connectivity incorporated into a network's architecture is often used to define its type (Franklin, 1989; Gallant, 1988). Fully-connected networks would involve the output of each and every neuron being connected to an input of every neuron including itself. Partially-connected networks would omit some of these connections. However two popular architectures arrange neurons in layers:

Layered feedforward networks (Ivakhnenko, 1971) process data values unidirectionally from inputs towards outputs via a number of intervening layers of neurons. These are referred to as hidden layers, because their outputs are not directly connected to (and accessible to) the outside world. The network's

inputs are connected to the first layer. Outputs of each successive layer are connected to inputs of the next layer. Between adjacent fully-connected layers, each and every output of the previous layer is connected to an input of each neuron on the subsequent layer. Such feedforward networks thus do not incorporate memory, the outputs of the network at time *(t)* being entirely determined by the network's inputs at time *(t)*, given the values of the weights and biases. Typically the number of neurons in the output layer is defined by the nature of the problem and the coding adopted for the solution. However the number of neurons on each of the hidden layers is a degree of freedom in the design of the network's architecture. Han et al. (2007) propose the rule of thumb:

$$N_{hu} = (N_{in} + N_{out}) \times 2/3 \qquad (2)$$

where: N_{hu} = Number of hidden neurons; N_{in} = Number of input neurons and N_{out} = number of output neurons. This was originally proposed in the PhD thesis of J Han (2003) on fluvial runoff modelling directly from rainfall radar reflectance images.

Recurrent networks (Elman, 1991, 1990; Grossberg, 1976) typically also arrange neurons in layers, but provide connections from outputs (at time t-Δt) of layers towards the output back to inputs (at time *(t)*) of layers nearer to the input and/or within each layer, providing the network with memory. Recurrent networks are not employed in this thesis, so are not discussed further here.

ANNs have been used for a wide variety of purposes in classification (Bartlett, 1998; Ultsch, 1993; Wang et al., 2010), regression (Sarle, 1994; Specht, 2006, 1991), control (Franklin, 1989; Kim On and Teo, 2010; Sontag, 1993), robotics (Kim On and Teo, 2010; Lewis et al., 1998; Miyamoto et al., 1988) and also data-driven modelling (Brion and Lingireddy, 2003; Dibike et al., 1999; Solomatine, 2008). It is this last application that is considered here.

Since the inception of Perceptrons (Rosenblatt, 1958) the idea of ANNs as function approximators that could be trained has existed. However, it is only through and since the development of Multi-Layer Perceptrons (MLPs) (Hornik et al., 1989; Rumelhart and McClelland, 1986a) that ANNs have become

accepted as universal function approximators. As Data-Driven Models (DDMs) they require calibration/training prior to their use in a predictive scenario. The kernel of the training algorithm invariably involves the adjustment of the multiplicative weights and biases at the inputs of each neuron in the network (White, 1989). However, training may also involve making changes to the architecture of the network, for example making/breaking connections and/or numbers of neurons in the network, especially of the hidden layer(s).

Regarding the function approximation capabilities of ANNs, Cybenko (1989) demonstrates that any decision region can be arbitrarily well approximated by continuous feedforward ANNs with only a single internal, hidden layer using any continuous sigmoidal nonlinearity (as its activation function). This is an important finding and as a result, research to date has been focused largely on 1HL networks. Acceptable solutions for many problems are obtained with these, including those described in the later chapters. Similarly, Barron (1993) also provides proof that a single layer of sigmoidal nonlinearities (1HL) can approximate any given function with an error $O(n^{-1})$ where n is the number of sigmoidal functions (units) in the layer. The techniques described in this thesis are applied to 1HL networks, but could be extended to deep networks (Ciresan et al., 2012; Collobert and Weston, 2008; Hinton et al., 2012) if desired.

1HL networks are also often designated "3-layer" feedforward ANNs. There are 2-layers of neurons; the extra (input) layer being merely a distribution mechanism for every input signal to connect with every neuron in the single hidden layer. In layered feedforward ANNs, each neuron implements the transfer function[2]:

$$y = f(x) = \kappa\left(\sum_i w_i g_i(x) + b\right) \qquad (3)$$

where: x is the input, $g_i(x)$ is some function of x, implemented by the neuron(s) towards the input of the network (for the input layer and hidden layer

[2] *This is a generalisation of equation (1) allowing for the cumulative effects of the layers towards the input being recursively defined in the function g(x)*

$g_i(x) = x$), w_i is a weight associated with input i, b is a time-invariant bias level and κ is an activation function applied to the output of the neuron. The hidden layer activation functions implement the sigmoidal nonlinearities referred to in Barron (1993). Figure 2.3 illustrates a feed-forward "3-layered" or "1HL" ANN, which is fully-connected between adjacent layers. This is an example of what Sjöberg (1995) refers to as a "NFIR" (Non-linear Finite Impulse Response) model.

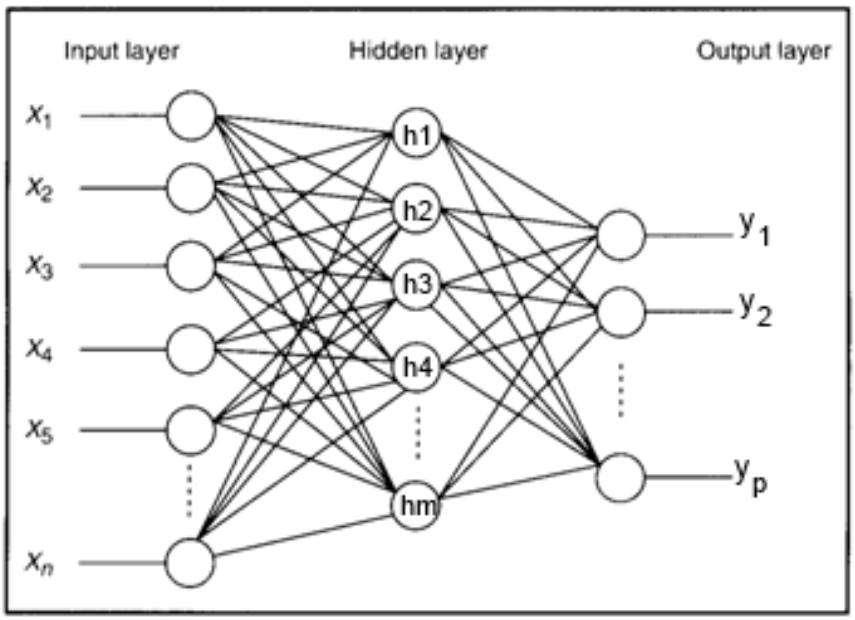


Figure 2.3. Three-layered feedforward ANN ("1HL") (MechanicalForex.com, 2014)

2.2.2 Supervised and unsupervised learning

Calibration of ANN-based and other DDM's is accomplished through a learning process referred to as training. In the case of ANNs, training involves adjusting the weights and biases of the network[3] to produce the desired behaviour of the trained network.

Machine learning model training algorithms are generally divided into two main types: supervised and unsupervised. In supervised learning (Jordan and Rumelhart, 1992), the expected target response ($t_i \in T$) that the network is to make to each input sample ($x_i \in X$) in the training dataset $\{X,T\}$ is known *a priori*. This information is provided by a notional "teacher"; hence the term, "supervised". Training proceeds by some means to minimise the error between the response the network makes ($y_i \in Y$) and the expected target response ($t_i \in$

[3] *Other features of the ANN's architecture may also be modified, dependent on network type.*

T). In unsupervised learning (Barlow, 1989; Figueiredo and Jain, 2002; Hofmann, 2001; Saul and Roweis, 2003), the target responses of the network are not known *a priori*. Instead the algorithm attempts to cluster the input data according to an appropriate rationale. In both cases the object is to use the training dataset to train the ANN model so as to be able to generalise a response to new data samples in a separate "test" dataset or, indeed, in a live real-time system. Throughout this thesis, supervised learning is used; therefore this discussion focuses on it.

2.2.3 Online and offline learning

A further categorisation of ANN training distinguishes between online (Liang et al., 2006) and offline (Langley, 1996) learning. In offline learning an entire training dataset is used to update the network's weights. Training progresses via a number of "epochs" or training steps. At each epoch, the entire training dataset is presented to the ANN and errors are computed for its response over that entire set of samples using a suitable metric. In online learning, weight updates occur upon each new sample being presented to the network. Effects of weight changes on the error are computed by evaluating differences (for both errors and weights) between the current and the previous timestep (right-hand side of equation (4)). Consequently the weight updates are computed (the left-hand side of the equation). This is an approximation, so the process progresses iteratively.

$$w_{ij}(t+1) - w_{ij}(t) \propto -\frac{\partial E(t)}{\partial w_{ij}(t)} \qquad (4)$$

where: $w_{ij}(t+1)$ is the weight associated with the connection from the *j*-th unit output of the previous layer to the *i*-th unit input of the current layer at the next timestep; *E* is the error and $\partial E(t)$ is the (partial derivative) change in error in the output between the previous and the current timestep; $\partial w_{ij}(t)$ is the (partial derivative) change in the same weight value between the previous and the current timestep; $\propto$ indicates "is proportional to".

A potential difficulty with supervised online learning in a live system is providing the target values in a timely manner. This particularly applies to supervised training of predictive models operating in real-time, where it would

be necessary to wait until the time of observation of the predicted phenomenon before a sample including observed target value could be added to the training dataset. An example of an online training algorithm (Yang and Amari, 1997) illustrates this potential difficulty and solves it using an unsupervised, clustering approach.

2.2.4 Feed Forward Back Propagation (FFBP)

This section discusses the FFBP approach to training (Hecht-Nielsen, 1989; Rumelhart and McClelland, 1986a). The development of this ANN training algorithm by Rumelhart, Hinton and Williams during the 1980's provides a reliable method of training for MLPs and therefore represents a significant step forward for machine learning. This is an example of supervised learning, since it is based on a set of known target output responses from which the error in the ANN's output can be propagated backwards and weight updates therefore computed.

The algorithm for FFBP has two phases: forward pass – in which the training data samples are presented to the network and outputs computed; and backward pass – in which errors are computed and propagated backwards. Algorithm 1 below defines the methodology:

Algorithm 1: Feedforward Backpropagation Algorithm (Rumelhart and McClelland, 1986a)

Input: training dataset with features **I** and instances **N**
Output: set of weights **W[*]** (biases treated as weights with fixed input of 1).

```
1:  set all weights W[*]← rand[*]
2:  repeat
3:     for every pattern in the training set
4:        Present the pattern to the network
5:        //       Propagate the input forward through the network:
6:        for each layer in the network
7:           for every node in the layer
8:              Calculate the weight sum of the inputs to the node
9:              Add the threshold to the sum
10:             Calculate the activation for the node
11:          end
12:       end
13:       //       Propagate the errors backward through the network
14:       for every node in the output layer
15:          calculate the error signal
16:       end
17:       for all hidden layers
18:          for every node in the layer
```

```
19:                 Calculate the node's signal error (compare with last epoch)
20:                 Update each node's weight in the network
21:             end
22:         end
23:         //      Calculate Global Error
24:         Calculate the Error Function
25:     end
26: while ((maximum  number of iterations < than specified) AND (Error Function is
    > than specified))
```

2.2.5 Gradient Descent (GD) versus Scaled Conjugate Gradients (SCG)

Gradient Descent (GD) is a generalisation of the FFBP algorithm. It is a popular method of optimisation of ANN weights and biases during training. For each layer, the partial-derivatives of the errors with respect to the weights are calculated (Jacobian matrix). Weight values are then updated in the direction of steepest descent from the current position in the weight space. For this reason the algorithms are known as gradient-descent methods. The Quasi-Newton (Battiti, 1992) method implements GD using an approximation for computation of the Jacobian matrix. Similarly, the Levenberg-Marquardt algorithm (Marquardt, 1963) uses the method of damped least squares iteratively to arrive at a minimum of error with respect to the weights. However this is not guaranteed to be a global minimum; depending on (randomised) initial values of the weights, local minima may also be discovered.

Conversely, the Scaled Conjugate Gradients (SCG) algorithm (Møller, 1993) only initially commences in the direction of steepest descent. After this, it computes a direction at each iteration, which allows it to converge to the error minimum in O(N) iterations, where N is the total number of weights and biases in the ANN. This compares with $O(N^2)$ iterations for GD. Møller reports at least an order of magnitude improvement in speed of convergence over GD for the test problems used in his paper. In order to achieve this, approximations for the second-order partial derivative of error with respect to the weights (Hessian matrix) and the optimal step-size at each iteration are employed in the SCG algorithm. In order to illustrate the difference between GD and SCG, Figure 2.4 shows a contour plot of error versus the values of (a trivial example ANN with only) two weights. Training commences at location x_0 (selected randomly), whereas the optimum is at x.

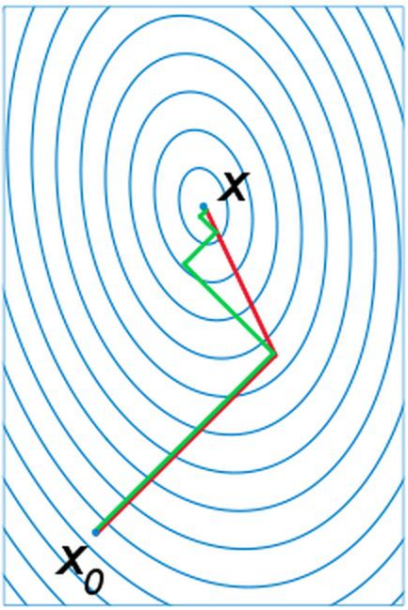


Figure 2.4. Comparison of GD with SCG for 2-weight ANN (Wikimedia Inc, 2015)

The green line shows the progress of a gradient descent algorithm, following the direction of steepest descent at each iteration; whereas the red line indicates the progress of the SCG algorithm, locating the optimum in only 2 steps; the same number as the number of ANN weights in this trivial case. It is worth noting that SCG does not take the direction of steepest descent on the second step (and potentially subsequent steps).

GD-based methods are often computationally expensive, since the error-versus-weight matrices may be of high rank, and very large. Some involve not only computation of first order (gradient) "Jacobian", but second order (curvature) "Hessian" matrices (Gradštejn, 2000). The SCG method avoids computation of the Hessian matrix by using an approximation using the Jacobian matrix instead. GD and SCG are single-threaded algorithms, so only explore a single region of the weight space at each iteration in the search for the global optimum. Devices such as learning rate and momentum have been added to variants of GD to try to overcome the potential for stagnation in subsidiary minima (Jacobs, 1988; Yu and Liu, 2002; Yu and Chen, 1997). Nonetheless, GD methods are popular and are used for many purposes. Where gradient-based ANN optimisation is required for case studies described in this thesis, the SCG algorithm is used due to its superior performance when compared with GD.

2.2.6 Approaches to Prevention of Overfitting

A problem that can arise during ANN training is known as “overfitting”. Hawkins (2004) defines it as follows:

“Overfitting is the use of models or procedures that violate parsimony: that is, that include more terms than are necessary or use more complicated approaches than are necessary.”

This could include use of models that are more flexible than they need to be (perhaps using non-linear terms, when a linear model would suffice; or incorporating irrelevant terms in the model – for example the use of independent variables that are irrelevant to prediction of the dependent variable. From a signal-processing perspective, if it is assumed that data samples contain both a “signal” element and a “noise” element, then overfitting equates to fitting the model to both the signal and the noise instead of just to the signal. The result of overfitting to a training data set is that the model does not generalise well to new “test” data resulting, for example, in sub-optimal prediction performance.

Popular methods aimed at prevention of overfitting include Early Stopping (Caruana et al., 2001), Weight Decay Regularisation (Hawkins, 2004; Moody et al., 1992, 1995) and the use of feature selection approaches detailed in section 2.6. One feature selection approach of particular significance to this thesis is Automatic Relevance Detection (ARD) (MacKay, 1995) described in section 2.6.3.

Early Stopping is a technique in which a set of samples from the training dataset are reserved for the purpose of validation of true training progress during periodic pauses in the training process. Because the validation samples are excluded from the training dataset, the ability of the model to generalise is tested by the validation process. As training progresses, the error between the model’s desired target response and its actual response tends to decrease. However, at a certain point, validation error may reach a minimum, before starting to increase again (vertical dotted line in Figure 2.5). At this point the

model's ability to generalise is optimal. The termination of training at this point is known as Early Stopping. Overfitting is thus to a large extent avoided.

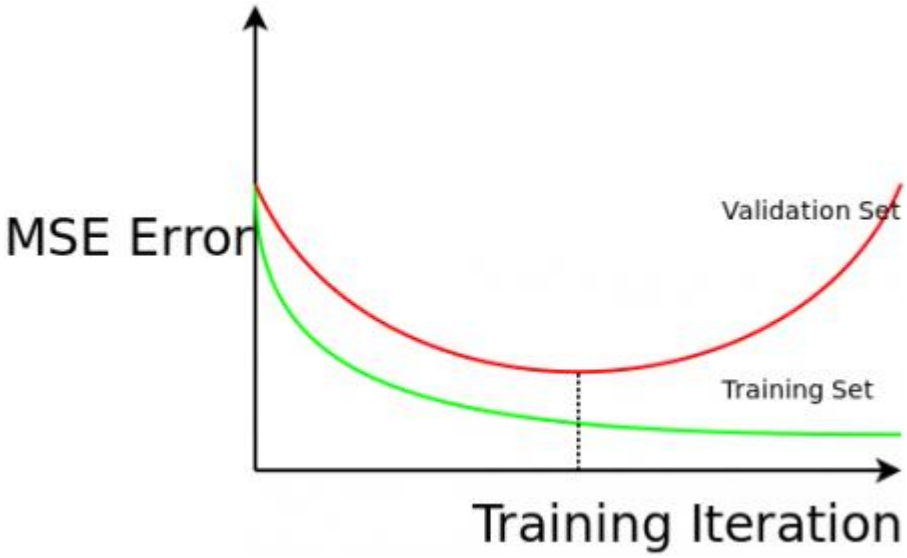


Figure 2.5. Validation Error used in Early Stopping (Larkworthy, 2013)

By contrast, Weight Decay Regularisation approaches avoidance of overfitting by arranging to penalize high weight values. Instead of using just the mean squared error (the left-hand term of equation *(5)*) as the performance function during training, an additional term is added to penalize the sum of the square of the network weights (including the biases):

$$E = \frac{1}{N}\left(\sum_{i=1}^{N}(T_i - Y_i)^2\right) + \alpha\|W\|^2 \qquad (5)$$

where: E is the training error; N is the number of samples in the training set; T_i is the target value for the i^{th} sample in the training set; Y_i is the model output value for the i^{th} sample in the training set; α is a weight decay constant s.t. $0 \leq \alpha \leq 1$ and $\|W^2\|$ is the L_2 (Euclidean) Norm of the network's weights.

The theory is that large values of weights permit fitting to the noise component of the data and thus lead to overfitting; so, if α is set to a value greater than zero during training, in order to minimise the error the sum of the square of the weights has also to be kept small. A trade-off will exist between the two terms and, if α is chosen appropriately an optimum of the ability of the model to generalise should be found.

Early stopping and weight decay regularisation are not incompatible with each other and several of the experiments described in chapters 3 to 5 use both in their efforts to ensure that overfitting does not occur.

2.2.7 Lagged Inputs and Moving Time Windows

A common method of handling time-series data with feedforward ANNs (which therefore have no inbuilt memory) is to use time-lagged inputs in a moving time window, when presenting data to the inputs of the network (May et al., 2008; Mounce et al., 2011). The effect of using such a window on an input data signal is similar to that proposed by Parzen (1962), which involves smoothing at different sample-scales. However, in this approach, unlike Parzen, there is no formal probability-density estimation; rather, the ANN uses the data to perform its own smoothing, optimised during training.

In signal-processing applications, the input data features are time-variant signals, in which each sample observation (instance) of the time-series has an implicit time sequence (timestep) associated with it. The timestep is normally of a constant size. Typically, due to lags within the real-world system being modelled, signal values from previous timesteps may influence (contain relevant information about) the system's output in the current timestep or future (predicted) timesteps. In urban and natural drainage catchments, lags are represented by the notion of Time of Concentration (ToC) (Butler and Davies, 2004), which is defined as the maximum travel time for water from any part of the catchment to the outfall. Any effective model must provide some means of modelling such systemic lags.

The moving time window, therefore, arranges to present in parallel a section or window of the time-series dataset X, $\{x_i(t-\tau), \ldots x_i(t-2), x_i(t-1), x_i(t)\}$ where: x_i is the *i-th* time-series signal (attribute) in X; t is the current timestep; τ is the maximum number of timesteps lag in the moving time window, for each of the signals *(i)* in X. The samples $\{x_i(t-\tau) \ldots x_i(t-2), x_i(t-1)\}$ are regarded as "lagged inputs" in the literature. The moving time window approach can be thought of as a data pre-processing step, which implements a shift-register or first-in-first-out (FIFO) buffer operating on each input time-series signal.

An interesting feature of this technique, when used with feedforward ANNs is that, once the parallelisation of the input data has been performed, each input sample remains fully independent of other input samples. So it would be legitimate to randomise the order of presentation of samples to the ANN, should this be desired. However, in hydroinformatics[4], "events", such as specific rain storms may frequently be modelled. This approach is used, specifically in the case studies of chapter 4 of this thesis. In these cases, it is desirable to preserve time sequence integrity of the observation samples within each event, so as to permit graphical inspection of the predicted hydrographs produced by the model output.

In the literature, the moving time-window, lagged input approach is often implemented (Bowden et al., 2005; Campolo, 2003; Fernando et al., 2005; Luk et al., 2000). However, the majority of studies only attempt to predict one timestep ahead. In operational terms this may be less than satisfactory, since timesteps may be very short, of the order of a few minutes. Our previous paper, Duncan et al. (2013b) investigates the limits of predictive capability for multi-nodal urban drainage networks, using a moving time-window approach with actual rainfall.

Using moving time windows, it is also not necessary or indeed optimal to use every timestep sample within the window. There is also the question of the optimal length of time window to use. Long time windows inevitably lead to a corresponding increase in the number of input features presented to the ANN, which increases the dimensionality of the decision space to be optimised during training. Methods for selecting which timesteps to use in the moving time window include Partial Mutual Information (PMI) (Luk et al., 2000; May et al., 2008) and cross-correlation (Fernando, 2005). These could be regarded as filter techniques for feature selection and others described in section 2.6.2 could equally potentially be applied. Similarly, wrapper-based approaches (described in section 2.6.1) could also be employed. Using cross-correlation assumes a linear relationship between input and output (target) variables, whereas use of mutual information (MI) or PMI allows for complex non-linear relationships between the two.

[4] *The case studies contained in chapters 5 and 6 are drawn from this field of research.*

2.2.8 Applications

As mentioned in the background section, ANNs are widely researched and used for applications across the fields of Pattern Recognition (Bishop, 1995, 2006), Control (Franklin, 1989; Kim On and Teo, 2010; Sontag, 1993), Signal Processing (Cochocki, 1993; Lapedes, 1987) and Predictive Modelling (Grayman et al., 2001; He et al., 2011; Liang and Liang, 2006) both for classification (Bartlett, 1998; Ultsch, 1993; Wang et al., 2010) and regression (Sarle, 1994; Specht, 2006, 1991) models. Classifiers predict the class label of each sample (such as "pass" or "fail" or flood categories "A", "B" or "C"); whereas regression models predict the (usually real) value of a quantity, such as water level or flood volume.

In Hydrology and the Environment, research is principally directed towards predictive modelling (Grayman et al., 2001; He et al., 2011; Herrera et al., 2010; Plumb et al., 2005), although their use for control (Verworn and Krämer, 2005; Zhang and Stanley, 1999) is also extensively studied. Applications also exist for pattern recognition, for example in the detection of locations of leaks in water distribution networks (Mounce et al., 2010, 2003) or in changes to the stability of coastal flood prevention dykes (Pyayt et al., 2011a, 2011b), although this overlaps with the area of predictive modelling.

Dawson and Wilby (2001) conduct an early review of research into ANN's use for rainfall-runoff modelling. They draw attention to the need for comparison of models and the general lack of a systematic approach to comparison up to the date of publication.

Abrahart et al. (2012) conduct an excellent survey of research to date in the area of use of ANNs for river forecasting, which is a problem akin to urban flood modelling and prediction. The paper also calls for more standardisation of datasets, to facilitate comparative study of modelling approaches and their effectiveness and accuracy. At the time of writing, this still appears to be a requirement.

Research into ANNs for urban flooding and related applications is reviewed in section 2.9.1 whereas ANN's and other machine learning

approaches to bathing water quality prediction is covered in section 2.9.1. These are given their own sections as they constitute the major research areas used for the case study chapters within this thesis.

2.3 Evolutionary Algorithms

Evolutionary Algorithms (EAs)(Whitley, 2001) are a popular nature-inspired approach to machine learning, search and optimisation. They are based on the principles of genetic coding, natural selection, recombination and mutation. There are many types of EA:

- Genetic algorithm (GA) - the most popular type of EA. the solution of a problem is sought in the form of strings of numbers (traditionally binary, although real number representations are also used)
- Genetic programming (GP) - solutions are in the form of computer programs such as model or decision trees; their fitness is determined by their ability to solve a computational problem.
- Evolutionary programming (EP) - similar to GP; however, where the structure of the program is fixed but its numerical parameters are allowed to evolve.
- Gene expression programming (GEP) - like GP, GEP also evolves computer programs but explores a genotype-phenotype system, where computer programs of different sizes are encoded in vector chromosomes of fixed length.
- Evolution strategy (ES) - works with vectors of real numbers as representations of solutions, and typically uses self-adaptive mutation rates.
- Differential evolution (DE) - based on vector differences; is therefore primarily suited for numerical optimization problems.
- Neuroevolution (NE) - similar to GP but the genomes represent artificial neural networks by describing structure and connection weights. The genome encoding can be direct or indirect.
- Learning classifier system (LCS) - here the solutions are classifiers (rules or conditions) that can include ANNs.

2.3.1 Definition and description of EA

An evolutionary algorithm (EA) is a generic population-based meta-heuristic optimisation algorithm. Candidate solutions to the problem to be optimised play the role of individuals in a population, and the objective (fitness) function determines the relative quality of each of the solutions. Evolution of the population then takes place after the repeated application of the selection, crossover and mutation operators. EAs often perform well in approximating solutions to all types of problems because they generally do not make any assumption about the underlying fitness landscape (unlike gradient-based algorithms). This means that they can operate even where the fitness landscape is non-differentiable. In most real applications of EAs, such as the training of ANNs (Neuroevolution), computational complexity can be a prohibiting factor. This computational complexity is largely due to objective function evaluation, which in the case of NE involves instantiation of an ANN and simulation using the whole training dataset for each and every member of the population of candidate solutions.

Algorithm 2 illustrates the most popular form of EA – the genetic algorithm (GA), which, in its real-valued form, is used in this thesis. β is the population of candidate solutions and t is the current generation.

Algorithm 2: Genetic Algorithm Source: (Ortiz-Boyer, 2005)

Input: Population of initial candidate solutions $\beta(0)$
Output: Population of optimised candidate solutions $\beta(t)$

```
1:  begin
2:      t ← 0
3:      initialise β(t)
4:      evaluate β(t)
5:      while (not stop_criterion) do
6:      begin
7:          t ← t + 1
8:          select β(t) from β(t - 1)
9:          crossover β(t)
10:         mutate β(t)
11:         evaluate β(t)
12:     end
13: end
```

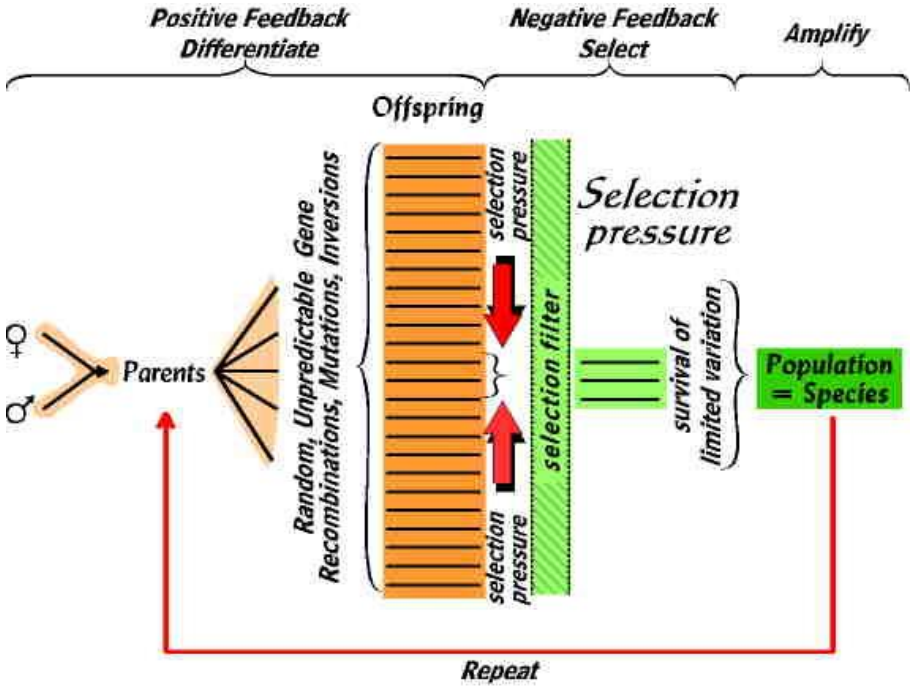


Figure 2.6. General Evolutionary Algorithm block diagram (JamesMadisonUniversity, 2012)

EAs are optimisation heuristics that maintain populations of candidate solutions and apply operators akin to genetic-crossover, mutation and natural selection and recombination on that population in order to generate fitter solutions in each generation. Figure 2.6 illustrates this in schematic form. The selection filter selects only the fitter solutions to become the parents for creating the offspring solutions for the next generation. In some GAs, the parents are also included in the next generation; in others, only the child solutions for the next generation. Thus the population evolves (typically asymptotically) towards an optimum of fitness in the so-called objective space. This is achieved by altering the values of coefficients in the decision space ("decision variables"). It is helpful also to think of EAs as search algorithms that explore the decision space at multiple locations simultaneously, hopefully to discover the global optimum in the objective space. However, this is not guaranteed with a finite population size. The potentially high-dimensional relationship between the decision space and the objective space is defined by the nature of the real-world problem being solved.

EAs fall into two categories Single Objective (SOEA) and Multi-Objective (MOEA) depending on the number of independent objective measures of fitness used to evaluate members of the population. These can be regarded as independent axes / dimensions in the objective space.

2.3.2 Operators

2.3.2.1 Selection

Selection is one of the three most important operator types used in EAs. It provides the means to drive the fitness of the population as a whole forward. A number of variations exist. In proportional selection, the individuals are selected with probability according to their relative fitness. Ranking selection assigns selection probabilities on the basis of the ranking order of individuals' fitness, ignoring absolute fitness values. Tournament selection is performed by choosing *q* parents randomly from the population and reproducing the best individual from this group. The most commonly implemented is binary tournament selection, where *q=2*. This is also known as elitism. Genitor selection (Whitley, 1989), is a steady-state selection method and works individual by individual. Each time, one individual is chosen according to linear ranking and then the worst individual in the population is replaced.

Zhang and Kim (2000) review a number of selection operators including proportional selection, ranking selection, linear ranking, tournament, Genitor selection. These are compared to algorithms based on simulated annealing, and hill-climbing. Four selection methods are compared: proportional, ranking, tournament and Genitor. The paper focuses on the practical problem of machine layout. It compares the quality of solutions obtained in a reasonable amount of time by each of the four. Mutation and crossover operators are also used. The study concludes that the methods of ranking and tournament selection obtain better results than the methods of proportional and Genitor selection – at least for the chosen problem.

2.3.2.2 Crossover

Crossover is the mechanism by which the chromosomes (of decision variables) in the parent solutions are combined to provide the new chromosomes of the offspring generation. Binary crossover is the most commonly used. Figure 2.7 illustrates the binary crossover process.

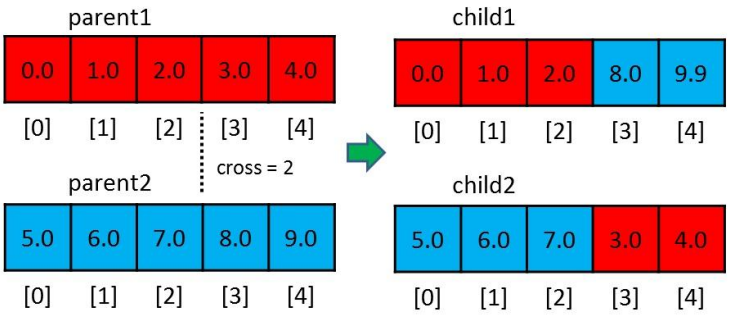


Figure 2.7. Crossover operator (McCaffrey, 2014)

On the left are the notional 2 chromosomes of the parents with 5 decision variables each. These form 5 base pairs. Indices for the bases are shown in square brackets. A crossover point of 2 has been previously chosen (i.e. after base pair index [2]). On the right are the two chromosomes after crossover. After the first crossover point (there may be several) the bases in each pair swap positions, up to the point of the second crossover (or the end of the chromosome is reached as in this case).

Other crossover operators also exist, such as cycle crossover (Oliver et al., 1987).

2.3.2.3 Mutation

Mutation is a mechanism by which diversity in the population is supported. It allows new regions of the decision space potentially to be explored by introducing new values for the decision variables as opposed to crossover, which explores decision space by re-using existing values in different combinations. Choice of appropriate mutation operator is very much dependent on the definition of the problem as well as its representation in terms of decision variables. The original GA proposed by Holland (1975) uses binary representation for the bases on the chromosome and mutation is performed by simple inversion of the bit(s) to be mutated.

Mutation operators include: displacement mutation, exchange mutation, insertion mutation, simple inversion mutation and scramble mutation. These are all reviewed in Larrañaga (1999). Displacement mutation involves cutting out a section of the chromosome, moving it along and re-inserting it. Exchange mutation involves swapping bases from different locations on the chromosome.

Insertion mutation is similar to displacement mutation except that only a single base is moved at a time. Scramble mutation involves the random selection of a section of chromosome and then randomly reassigning the order of the bases within it.

For real-valued representations of the chromosome, options include replacement mutation (replacing a base with a new random value) and incremental mutation (altering the value of a base by a random amount (within a certain value range of the existing value). This range can also be varied with time as the optimisation progresses. Usually, it is reduced so that near-optimal regions of the decision space are explored more finely as the algorithm progresses (Michalewicz, 1996).

2.3.3 Multi-Objective EAs

Much work has been carried out on using Multi Objective Evolutionary Algorithms (MOEAs) (Zhou et al., 2011). These involve more than one objective function; all requiring to be optimised simultaneously. These objectives usually exist in a trade-off with each other; thus improving one objective may lead to the degradation of the others. In this case the objective functions are said to be conflicting and there exists a large or even infinite number of Pareto optimal solutions. A solution is called "non-dominated" or "Pareto optimal" if none of the objective functions can be improved in value without degrading at least one of the other objective values. Without additional subjective preference information, all Pareto optimal solutions are considered equally good. This is due to the fact that it is not possible to order vectors completely. Usually a human decision maker (DM) is required to apply personal preferences and select the finally preferred non-dominated solution.

The definition of non-domination, from Jin et al. (2009) follows:

Formally, consider the following multi-objective minimisation problem:

Minimise	$f_m(\boldsymbol{x})$	$m = 1, 2, \ldots M;$
subject to	$g_j(\boldsymbol{x}) \geq 0,$	$j = 1, 2, \ldots J;$
	$h_k(\boldsymbol{x}) = 0,$	$k = 1, 2, \ldots K;$
	$x^L_i \leq x_i \leq x^U_i,$	$i = 1, 2, \ldots n;$

where $f_m(\boldsymbol{x})$ are the M different objective functions to be minimized,

$\boldsymbol{x} = (x_1, x_2, \ldots x_n)^T$ is the n-dimensional decision space,

$g_j(\boldsymbol{x})$ are the J inequality constraints,

$h_k(\boldsymbol{x})$ are the K equality constraints, and

x^L_i and x^U_i are the lower and upper bounds of the *i-th* decision parameter, respectively.

For the multi-objective minimization problem defined above, solution $\boldsymbol{x}^{(1)}$ is said to dominate solution $\boldsymbol{x}^{(2)}$, if $\boldsymbol{x}^{(1)}$ is no worse than $\boldsymbol{x}^{(2)}$ in all objectives, i.e.,

$\forall m = 1, 2, \ldots M, \quad f_m(\boldsymbol{x}^{(1)}) \leq f_m(\boldsymbol{x}^{(2)})$, and

if $\boldsymbol{x}^{(1)}$ is strictly better than $\boldsymbol{x}^{(2)}$ in at least one objective:

$\exists m' \in \{1, 2, \ldots M\}$, such that $f_{m'}(\boldsymbol{x}^{(1)}) < f_{m'}(\boldsymbol{x}^{(2)})$.

2.3.3.1 NSGA-II algorithm

The nondominated sorting genetic algorithm II (NSGA-II) (Deb et al., 2002b) has become an academic and industry standard MOEA. The authors claim that it is a fast algorithm due to its computational complexity $O(MN^2)$, where M is the number of objectives and N is the population size. This is an improvement on the previous level of $O(MN^3)$ for GAs. It is also elitist, since the selected parents (the fitter half of the previous generation – based on binary tournament selection) are also included in the population for the new generation. It is a flexible algorithm, capable of working with various codings of the problem – reflected in the values used for the chromosome of decision variables (e.g. binary, integer or real). The kernel of the algorithm sorts the solutions into ranks, where rank 1 solutions are nondominated, rank 2 are dominated by 1 solution each and so on. For solutions of equal rank, selection is also based on crowding distance, so as to achieve the best possible spread of solutions along the Pareto front in the M dimensional objective space. The algorithm itself is stated in chapter 5, where it is used in the case study experiments.

2.3.4 Applications

NSGA-II is very widely applied including to energy generation expansion planning (GEP)(Kannan et al., 2009); chemical reaction engineering (Nandasana et al., 2003); vehicle routing (Jozefowiez et al., 2006) and the dispatch problem (Dhanalakshmi et al., 2011).

In hydrology and the environment it is for example applied to the calibration of a catchment soil and water assessment tool (SWAT) (Bekele and Nicklow, 2007); optimisation of the design of water distribution networks (WDN) (Atiquzzaman et al., 2006; di Pierro et al., 2009; Khu and Keedwell, 2005; Preis and Ostfeld, 2006); calibration of rainfall-runoff models (Couckuyt et al., 2009; Khu and Madsen, 2005, 2003; Nazemi et al., 2008, 2006); optimisation (both of design and control) of wastewater treatment processes (Beraud et al., 2009; Fu et al., 2009, 2008; Iqbal and Guria, 2009); optimisation of flood management solutions (Delelegn et al., 2011) and reservoir modelling and control (Kim et al., 2008; Kim and Heo, 2006). In addition, in studies too numerous to cite, it is used as a benchmark for comparison with novel algorithms and techniques.

2.4 EAs as a Method for Training ANNs (Neuroevolution)

Evolutionary Algorithms (EA) – specifically real-valued or mixed-valued Genetic Algorithms (GA) (Goldberg and Holland, 1988) can be used to evolve ANNs. See for example Yao (1993), Branke (1995) and the review paper by Yao (1999) and the state-of-art review by Zhou et al. (2011). These involve using various versions of single or multi-objective EAs to optimise the values of weights, network architectures and/or learning rules of ANNs.

In the case of training ANNs the EA's decision variables are the values of the ANN weights and biases (and optionally other architectural configuration parameters). The number of weights and biases may of course also vary depending on decisions about the architecture of the ANN; such as the number of hidden units. Depending on the algorithm used, diversity within the population can be arranged against a number of criteria; for example use of crowding-distance in NSGA-II (Deb et al., 2002b) combined with definition of objectives in the fitness function that measure factors such as architectural-complexity or

weight-regularization; or use of correlation penalty (Liu and Yao, 1999a, 1999b) in the fitness function (evaluated across the whole population of ANNs together during simultaneous training of all members).

2.4.1 Review of SOEAs in this field

Single-objective evolutionary algorithms (SOEAs) are less frequently used in the literature for the training of ANNs than their multi-objective counterparts. However, where a single objective is used, for example in the Genitor algorithm (Whitley et al., 1990), a single metric is evaluated; here the sum of squared error (SSE), evaluated when applying the whole training dataset. Binary coding is used on the chromosome, corresponding to the ANN weights used to solve simple binary coded problems:

1. the exclusive-or (XOR) problem,
2. a 424-encoder, and
3. two versions of an adder problem

Thus, in this case, the scenario of evolutionary optimisation of real-coded ANN weights and biases is avoided.

Guo and Uhrig (1992) use a GA to select input features to modular ANNs designed to classify various fault modes in a nuclear power plant. The input features are selected based on three criteria that are combined into a single fitness objective function. The three criteria are:

1. Number of inputs
2. % error on the classification
3. Generation index of the GA

It would have been possible to treat each of these as separate objectives, since it is likely that trade-off relationships would exist between these variables. However, by combining into a single objective, the search algorithm is simplified and the decision about which ANN from the final population to use is reduced to selecting the one with the highest level of fitness. A disadvantage of the method is that two runs of the algorithm with the same inputs would be unlikely to result

in the same ANN solution (and hence same % error of classification), due to the different random initialisations of the weights.

2.4.2 Review of MOEAs in this field

When training ANNs, additional objectives can be added such as a regularisation term (e.g. sum of square of the weight values), which serves to keep weight values as low as possible. This has been shown to reduce the probability of overfitting to the training dataset and thus reducing the ability of the network to generalise to new "test" data samples following completion of training (Jin et al., 2004).

Training the ANN thus becomes a multi-objective optimisation (MOO) problem, in which typically there is a trade-off between all of the objectives. A Pareto front of non-dominated solutions is thus developed. This is discussed in the literature including: (Hung and Chan, 2013; Jin et al., 2009; Kim On and Teo, 2010) and the definition of non-domination is stated in section 2.3.3.

Although many popular standard evolutionary algorithms exist (see below), these invariably require tailoring to the particular problem being solved. This is achieved by means of the Objective Function (OF) defined by the applications programmer. The EA invokes the objective function every time it needs to evaluate the performance of a candidate solution in the population. Internally, the OF applies the current value of the chromosome for the candidate solution and evaluates the performance of the solution according to a suitable metric (or metrics in the case of MOO). For ANN training, applying the chromosome means setting up the ANN architecture and the values of weights and biases. The entire training input dataset is presented to the ANN sample-by-sample and its response to each is collected. This is then evaluated as a whole using for example mean-squared error (MSE) or Nash-Sutcliffe Efficiency Coefficient (NSEC) (Moriasi et al., 2007; Nash and Sutcliffe, 1970) as an objective. Other objectives may also be evaluated in the OF.

At the end of a training run, a set of non-dominated solution ANNs exists. Purely from the perspective of the set of objectives, each solution is equally good. Therefore an additional set of preferences needs to be applied in order to

select the final “best ANN” solution from the set. Traditionally, this is the role of a human Decision Maker (DM).

2.4.3 Some critical analysis of MOEAs

Whitley (2001) provides a review of practical issues and common pitfalls in the use of EAs for solving a number of types of problem in optimisation and machine learning; a number of selection strategies are also discussed. One of the main challenges is the encoding of a problem for use by an EA. Whitley (1990) uses a binary coding for ANN weight values, whereas Yao et. al. (1999) also evaluate the benefits of direct real-value coding for ANN weights. Another practical issue in the case of ANN training is that it may be desirable to allow the ANN architecture (e.g. number of hidden units, activation function types, connectivity etc.) to vary. This will have a direct effect on the number of weights and biases to be set and hence the length of chromosome (dimensionality of the decision space). This may thus be different for different members of the population. The process of crossover is complicated, since this normally assumes the chromosomes of the two parent solutions are of the same length. Solutions to this have been proposed by Yao et. al. (1999). These include use of indirect coding of the ANN; instead coding a parametric representation of the ANN or developmental rule representation – effectively recipes for the creation of ANNs coded on the chromosome of the EA.

Yao et al. (1999) also propose a hybrid approach in which the multi-threaded broad weight-space search capability of EAs is combined with local search around the fittest evolutionary solutions using gradient descent, so as to fine tune weight values in proximity to the global optimum.

Where ANNs have multiple output nodes (as could be the case for example when simultaneously modelling many sewer nodes for urban flooding, using a 1:1 correspondence between ANN output nodes and sewer nodes) it is generally not feasible to evaluate the fitness of each output as a separate objective. This would be particularly true, where domination-based algorithms such as NSGA-II (Deb et al., 2002b) are employed. Performance of (for example) NSGA-II has been shown to deteriorate rapidly where more than 3-objectives are used (Ishibuchi et al., 2008). As the dimensionality of the

objective space increases, a situation is quickly reached where the vast majority of (or all) solutions in the population are non-dominated. This leads to a reduction in selection pressure, since all non-dominated solutions would be selectable as parents in the tournament used in NSGA-II.

EAs rely on maintaining diversity within the population in order to continue to provide selection pressure, in other words to improve the probability of breeding new solutions in the child generation that are fitter than the fittest solution in the parent generation. The mutation operator plays a key role in this, since it introduces new values into the genome allowing new areas of the decision space to be explored. Crossover also plays a role in exploration of new spaces, but only regions that can be reached by permutations of the same set of values within the decision space.

Typically tens or hundreds of sewer nodes are required to be modelled in urban flood prediction applications. Therefore metrics that aggregate ANN performance results across all output nodes together must be employed. It is possible that novel many-objective algorithms, such as HypE (Bader and Zitzler, 2010), may provide means for evolutionary training of multi-output ANNs, where the performance of each output is able to be treated as a separate objective. HypE uses the negative effect on a hypervolume metric of the entire Pareto-front that the removal of each single candidate solution would have, in order to evaluate the fitness of each individual solution in the population. However, to the author's knowledge this remains to be tested for training multi-output ANNs.

2.5 Cross-Validation Techniques

Cross-validation is an important tool in machine learning and data-driven modelling, since it provides a means of confirming performance of a model or models multiple times. Statistics for the mean and spread of performance lends increased confidence as to the robustness and repeatability of the model(s).

Usually, cross-validation involves division of the dataset using some rationale. Division is by observations (samples) rather than by features or attributes. These include N-fold cross validation (NFCV) and leave-one-out cross validation (LOOCV):

2.5.1 N-fold cross validation (NFCV)

Division of datasets into folds is a commonly applied approach in machine learning in order to perform cross-validation of models (Cawley and Talbot, 2003; Hansen and Salamon, 1990; Kohavi, 1995; Tiwari and Chatterjee, 2010a)[5]. It has also been used to create model ensembles. Shen et al. (2012), for example, use K-fold partitioning of a dataset to create an ensemble of models optimised using the Harmony Search algorithm (Lee and Geem, 2005).

2.5.2 LOOCV as a special case of NFCV

Leave-One-Out-Cross-Validation (LOOCV) is a standard methodology for dividing datasets for machine learning trials (Cawley and Talbot, 2003), in which a single sample is omitted from the training set and used afterward to test the trained model. This is repeated, using each and every sample in turn as the "left out" test sample. In this way, an ensemble of models with the same number of members as observation samples can be constructed. This is the extreme case of the application of the N-fold cross-validation (NFCV) approach[6]. Research has shown that the LOOCV level of data division is unnecessarily computationally expensive and does not lead to performance advantages over the use of a smaller number of data folds with several or many samples in each (Kohavi, 1995).

The LOOCV principle can be applied to the case where a smaller number (N) of multi-sample data folds are used – “N-fold Cross-validation” (NFCV). Observation data are divided (using some rationale) into distinct folds, each ideally containing a similar number of observation samples. In the case of urban flooding each fold might correspond to a rainfall event; whilst for bathing water quality, it might be a complete bathing season of compliance samples. Each fold in turn is used as the "leave-one-fold-out" fold – in the sense that it is excluded from the dataset used for training and validation of the model and instead used to evaluate its "test" performance after the training is completed.

---

[5] *In the limiting case of Leave-One-Out-Cross-Validation (LOOCV), a single sample can be omitted from the training set and used to test the model. This can be repeated using each and every sample in turn as the "left out" test sample. In this way, an ensemble of models with the same number of members as observation samples would be constructed.*

[6] *This is sometimes also referred to as K-fold cross-validation in the literature.*

In this way, each fold is used once to evaluate model test performance following training. It is also used once (in a different model) to check progress of performance during training – for example in an early-stopping regime to prevent over-fitting[7]. In each case, the remaining folds are used for training the model. The test results from all models are then typically aggregated to produce an overall measure of model test performance over the entire dataset; for example by taking a mean value.

2.6 Feature selection and extraction

A novel approach to feature selection is developed in this thesis as a method of model improvement based on the weights learnt during training of an ensemble of ANN models. Therefore it is necessary first to review feature selection methods in general. Feature selection is an approach to dimension-reduction of a machine-learning model's input space. It leads to improved parsimony and may lead to improved performance of models.

A number of differing approaches have been taken. Features can either be extracted from raw data in a dimension-reduction process (Feature Extraction) or they may be input signals used directly (Feature Selection).

Given a total number of available input features, *N*, each feature can either be included or excluded from the selected set of features; thus there are 2^N-1 possible combinations of input features (assuming at least 1 input is required). If *N* is relatively small, it becomes feasible (if not computationally efficient!) to conduct an exhaustive search for the best reduced input feature subset, known as a "reduct" (Wroblewski, 1995).

A paper by Liu and Yu (2005) reviews feature selection methods and attempts a meta-algorithm that integrates several approaches together in order to gain from the strengths of each. Feature selection strategies can be broadly divided into filter-based and wrapper-based approaches. The filter approach is applied as a pre-processor, whilst the wrapper approach embeds the machine-learning algorithm within the feature-selection process. Alternatively it can be

[7] *Checking of progress during training, using a separate data-fold excluded from the training dataset is also known as "validation". This should not be confused with N-fold cross-validation. The sense here is the validation of the training process itself rather than cross-validation of the final models across the whole dataset.*

viewed that the wrapper is similar to the filter approach apart from the fact that a learning algorithm is employed in place of an evaluation metric as used in the filter approach. Hybrids of the two also exist.

2.6.1 Wrapper-based approaches

Wrapper approaches to feature selection use the properties of the model itself (referred to as the 'inducer') to estimate an approximately optimal subset of input features. The novel approach described in Chapter 4 of this thesis and used in a case study in chapter 5 is a wrapper approach that encapsulates an ensemble of models.

Conventional greedy search strategies can be included under wrapper-based approaches. They are reported to be computationally advantageous and robust against overfitting (Guyon and Elisseeff, 2003). They may be categorised as either forward selection or backward elimination. For forward selection, input features are successively incorporated into larger and larger subsets Features are never removed, once added, hence the appellation “greedy”. Conversely in backward elimination the algorithm starts with the set of all features and progressively eliminates the least promising ones. Once eliminated, features are never added back. In both cases, these are wrapper based because the performance of the model itself is evaluated to assess the benefit of adding / removing each feature.

Kohavi and John (1996) describe a method of feature selection that classifies the 'relevance' of independent features into 3 classes; strongly relevant (correlated with target class; uncorrelated with other input features); weakly relevant (correlated with target class; correlated with at least 1 other input feature); irrelevant (uncorrelated with target class). The architecture of this 'wrapper' approach is shown in Figure 2.8.

The feature selection search attempts to use all strongly relevant features; (ideally) one weakly relevant feature from each sub-set of correlated weakly relevant features and reject all of the irrelevant features. They point out that relevance does not imply optimality and *vice versa*. For this reason, the performance of an inducer using the feature subsets is checked within the

algorithm (hence the 'wrapper' nomenclature) and finally an optimal feature subset is chosen for the induction algorithm. This is then evaluated as usual on new "test set" data. This method, although wrapping the inducer model, treats it as a black box. It is embedded in the algorithm in order to check performance during the feature selection process. The selection of features is performed based on analysis of correlations between the features themselves and between the features and the target class labels, rather than by analysis of parameters of the models themselves.

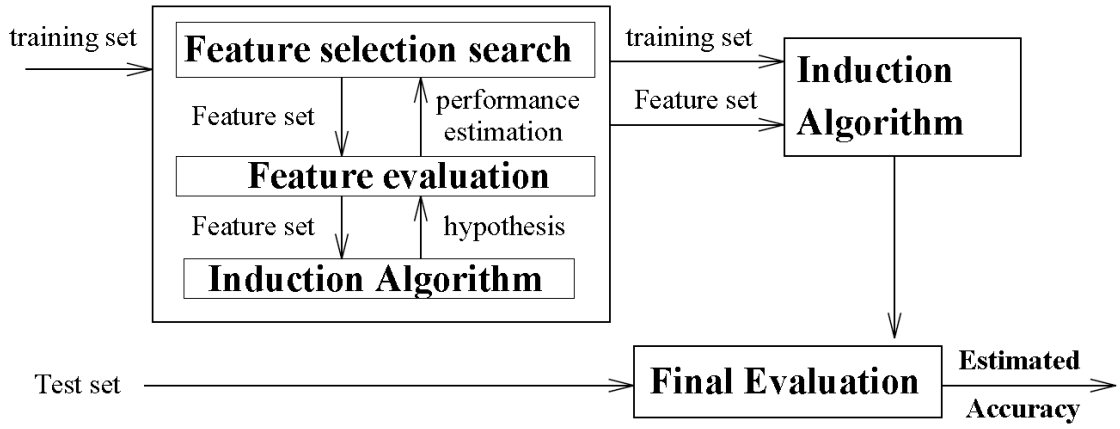


Figure 2.8. Wrapper approach to Feature Selection (Kohavi and John, 1996)

2.6.2 Filter-based approaches

Generally, the filter-based approach is used as a pre-processing step on learning datasets prior to their use and is independent of any learning algorithm that may be subsequently employed. Popular methods include fuzzy-rough feature selection (Jensen and Shen, 2009), probabilistic consistency based feature selection (Dash and Liu, 2003), and correlation-based feature sub-set selection (Hall, 1999). Individual feature-based methods are often also classified as filter-based. These typically have employed strategies such as hill-climbing where randomly selected additional features are added one at a time, until no further improvement in performance is achieved. However, this approach can lead to stagnation in subsidiary maxima rather than discovery of the true global optimum. Hybrid approaches (see section 2.6.3) such as random search or heuristic strategies have therefore been employed in order to try to avoid such shortcomings.

2.6.2.1 RELIEF / RELIEF-F

Kira and Rendell (1992a, 1992b) developed an algorithm (RELIEF) which can estimate the relevance of attributes using an information gain strategy applied prior to use of the resultant feature-subset by a machine learning algorithm. The algorithm is stated below:

Algorithm 3: RELIEF Feature Selection Algorithm

Input: training dataset with attributes **A** and **N** instances; ***n***: user-defined number of training instances to select
Output: set of weights **W[A]**.

1: *set all weights* **W[A]**← 0
2: **for *i*** := 1 to ***n*** **do**
3: **begin**
4: *randomly select an instance* **R**;
5: *find nearest hit* **H** *and nearest miss* **M**;[8]
6: **for A** := 1 to ***#all_attributes*** **do**
7: **W[A]**← **W[A] – diff(A,R,H)/*n* + diff(A,R,M)/*n***;
8: **end;**

Once executed, attributes (features) *A* can be selected based on *W[A]>T*, where *T* is a threshold, typically 0. They test RELIEF using parity problems of differing degrees with a significant number of additional input features containing random samples. RELIEF estimates probabilities for the classes in localised decision space. It is able to estimate the relevance of all features correctly in a time of the order of the number of features and the square of the number of training samples. Unlike Naïve-Bayesian (Kononenko, 1990) approaches, it can also take into account conditional dependencies between any of the independent variables. The algorithm can handle discrete and continuous features but is unable to handle incomplete data and is limited to two-class problems only. RELIEF has also been described as myopic, since it necessarily focuses only on nearest neighbours in each of the classes.

[8] *A "hit" is defined as a correct classification when compared to the target label for the sample (i.e. true positive or true negative) and a "miss" is an incorrect classification (i.e. false positive or false negative).*

Kononenko et al. (1994; 1997) extend this approach to address the above limitations and also to integrate the feature-selection approach with the machine learning algorithm. This they refer to as RELIEF-F. It is based on estimating probabilities more accurately than the original RELIEF by averaging results for k=10 nearest neighbours of each class instead of using a single nearest neighbour. For missing attributes (for any given sample) probable values are approximated from the relative frequencies for the classes in the whole dataset. To handle multiple classes *(C>=3)*, RELIEF-F evaluates probabilities for *k* nearest misses from each class (other than the "hit" class in each case). RELIEF-F outperformed a number of other methods including Naïve Bayes and *k-NN* classifiers, when tested on datasets designed to test the above types of problem. However, when tested on the UCI standard medical machine-learning datasets (Bache and Lichman, 2013) as well as others, Naïve Bayes and *k-NN* classifiers performed marginally better.

A study by Yang et al. (2011) into siting of Sustainable Flood Retention Basins (SFRBs) in Scotland uses 3 feature selection techniques together: RELIEF, Information Gain and Mutual Information and successfully reduces a set of 40 input features down to 9 relevant ones. They then successfully compare four benchmark classifiers (Support Vector Machine, K-Nearest Neighbours, C4.5 Decision Tree and Naïve Bayes) to verify the effectiveness of the classification with the pre-selected variables and automatically confirm the optimal number of variables. Case studies using 6-types of SFRB showed that the selected 9 relevant features were effective predictors for the 6 classes.

2.6.2.2 Principal component analysis (PCA) for feature extraction

Principal Component Analysis (PCA) (Jolliffe, 2005; Wold et al., 1987) is a broadly applicable approach to data dimension-reduction. It is a statistical technique that can be used for feature extraction in datasets with multiple input features. In this context, it fits broadly into the class of filter-based feature-selection techniques since it is a data pre-processing process, prior to model construction. It uses orthogonal transformation to convert a set of observations, in which input variables (features) may possibly be correlated with each other, into a set of values of linearly uncorrelated variables. These transformed variables are called principal components. In the case where correlation

between the original input variables is present, there will usually be fewer principal components than the number of original variables. The PCA transformation is defined using a ranking scheme so that the first principal component accounts for the largest possible amount of the variability in the dataset. Each successive component (in the ranking scheme) has the highest variance possible whilst also being orthogonal to (or in other words uncorrelated with) the preceding principal components. In this way, it is also possible to pre-determine a desired number of transformed input features (principal components) and simply to reject the components that are ranked as being less important than this limit.

Specialised ANN architectures have been developed to solve the PCA data transformation (Oja, 1989, 1992) but these could be viewed as a data pre-processing tool for a downstream learning task using the uncorrelated principal components (or a reduct of them) as inputs. Becker (1991) reviews PCA techniques in relation to ANNs in section 4 of her paper and points out that use of (orthogonal) principal components as ANN inputs can potentially simplify and accelerate gradient-descent-based training. This is due to the Hessian matrix being more nearly diagonal, yielding a simpler hyperplane in which to discover the region of steepest descent. Hsieh (2001) used specialised ANN architectures successfully to perform non-linear PCA using climate datasets from the Pacific Ocean to demonstrate their effectiveness.

2.6.3 Hybrid wrapper-filter approaches

Some feature selection methodologies use a combination of both the filter and the wrapper approaches. These include nature-inspired heuristics such as genetic algorithms (GA) (Wroblewski, 1995), Genetic Programming (GP) (Muni et al., 2006), Tabu Search (Hedar et al., 2008), and Particle Swarm Optimisation (PSO) (Wang et al., 2007).

A hybrid wrapper-filter ANN-based approach using "Random Permutation of Probabilistic Outputs" (Yang et al., 2009) has been used for feature-selection, in which a separate output neuron (with softmax activation function) is provided for each of c classes in the dataset. Each output estimates the probability of membership of the given class. As per standard practice, the class label for

each sample is predicted by the index of the output having the highest probability value. The approach uses a feature ranking criterion to measure the relative importance of each feature by computing the aggregate difference, over the feature space, of the probabilistic outputs of the neural network with and without the feature. Features are effectively removed in turn by randomly permuting their values between samples, whilst leaving all other features unchanged. The feature ranking criterion would ideally evaluate the aggregate value, over the entire feature space, of the absolute difference of the posterior probability $p(\omega_i|x)$ over all classes ω_i with and without a given feature x. However, a heuristic approximation is used in order to reduce computational complexity. Both filter and wrapper-based feature selectors are used. Filter-based techniques include Fisher-Score (FiS) and Mutual Information (MutI); whilst the wrapper-based technique of Maximum Output Information (MOI) (Sindhwani et al., 2004) is evaluated. This algorithm evaluates and seeks to maximise the mutual information between class labels and the output of the classifier, by using this as the objective function. It is computationally efficient, since it only uses discrete-valued quantities (from the confusion matrix of class labels) to evaluate MOI.

Zhu et al. (2007) presents a novel hybrid wrapper/filter feature selection algorithm for classification problems using a "memetic" framework (Krasnogor and Smith, 2005). This allows for individuals within a (phylogenetically learning) evolutionary population to perform local search, analogous to exploitation of ontogenetic learning of "memes" (Dawkins, 2006). The algorithm first uses filtering to identify core features based on improvement of performance of individual solutions when those features are added; then employs crossover, mutation and selection in the usual way to produce reducts, using a feature ranking approach. In the wrapper section of the algorithm, the EA at each evolutionary generation employs local search on all or parts of individuals' chromosome to create the next generation of improved solutions.

Automatic Relevance Determination (ARD) (MacKay, 1995; MacKay and Neal, 1994) uses a Bayesian framework that constructs predictive models, selects relevant features and rejects irrelevant ones. It provides a separate weight decay factor (α_i) for each input feature and adjusts them during the

Bayesian network training process, so as to minimise the values of α_i for irrelevant inputs – effectively turning them off. Penny and Roberts (1999) extend this methodology to “hard-ARD”, in which the irrelevant inputs are pruned from the model altogether in a second stage. This has the benefit of reducing the model complexity. The significance of ARD from the perspective of this thesis is that ARD provides a (Bayesian) alternative approach to automated feature selection that can effectively be regarded as using an ensemble of models. The approach in this thesis relies on an ensemble of ANN models too.

2.7 Ensemble Creation Techniques

Ensemble methods are applied across a huge range of disciplines. The discussion below limits itself to those methods commonly employed with ANNs and other related machine learning methods.

The advantage of an ensemble of models over individual models lies in the diversity found between the models in the ensemble. Each model is a “good” yet different model that makes a slightly different prediction in response to each sample within the dataset. It thus becomes possible to aggregate the predictions using some rationale, such as taking the mean of the ensemble in the case of regression models, or a majority voting decision in the case of classifiers. There are a number of approaches to creating ensembles with an element of diversity between the members:

2.7.1 Based on NFCV/LOOCV

Using NFCV, the diversity is created due to each model being trained on a different (even if overlapping) dataset of training samples. Dietterich (2000) uses 10-fold cross-validation to produce an ensemble of ANNs.

NFCV approach can be used for the generation of ensembles of ANN models in a way that is generally applicable, but particularly useful for time-series prediction (Dorffner, 1996; Khashei and Bijari, 2011; Lapedes, 1987) where the task is to create an auto-regressive model of order *p* "AR[p]", where *p* is the number of timesteps (signal samples) in a moving time-window of inputs. The use of a moving time-window of lagged inputs means that composite "samples" have to be assembled from a contiguous sequence of observations

(within the lagged time window) so as to apply them concurrently in parallel to the inputs of the ANN. It is therefore helpful (if not strictly necessary) to maintain the sequence of samples within each fold of observations or "event". If this is decided upon, approaches that employ random selection of samples are precluded. However, this is a special case and sometimes the randomisation of sample sequence before N-fold division is helpful. Provided the randomisation of sample order is done following the parallelisation of the time-lagged inputs, the order-independence of the samples is preserved and it once again becomes possible to randomise sample order if required; perhaps by methods such as bagging and boosting.

2.7.2 Based on Bootstrap Aggregation (Bagging)

A commonly applied technique is bootstrap aggregation or "bagging" (Breiman, 1996) in which a learning dataset L = $\{(y_n, x_n), n = 1 . . . . , N\}$ (where y_n is the *n-th* target output sample and x_n is the *n-th* corresponding input sample) is used to generate a sequence of *k* learning sets $\{L_k\}$. Each member learning set in the sequence is derived by sampling L with replacement (say) $K (<N)$ times at random. Thus each member learning set may have multiple *m* copies of any given sample s.t. $m \epsilon [0,1,2...K]$. Individual ensemble members can then be trained on each of the *k* learning sets in $\{L_k\}$. Breiman (1996) shows that substantial gains in accuracy (when compared with an individual model) can be achieved for ensembles of both classifier and regressor models using bagging. For example, Yu et al. (2008) use bagging to create a predictive model for credit risk estimation based on a number of predictive factors.

2.7.3 Based on Boosting

Boosting (Freund, 1995; Schapire, 1990) is a technique applicable to classifiers (and occasionally regressors) that allows weak learners to be combined together to produce a stronger group learner. This is as a solution to a question posed originally by Michael Kearns (1988). Schapire (1990) uses a recursive approach to combining the classifications of individual weak learners; whereas Freund (1995) simplifies this by taking a majority decision from the group as a whole. Training datasets for each weak learner are sampled subsets from the original complete dataset, using a so-called “distribution-free”

approach. Successive weak classifiers are made to focus on samples previously misclassified by the other weak learners by means of a sample weighting scheme.

In many machine-learning problems, datasets can be asymmetric. That is to say that numbers of samples attributed to each target class can vary widely, with some classes being far more frequently represented than others. In bagging, datasets are resampled randomly, which approximately preserves the original distribution of the target classes. In boosting, sampling from each class is performed according to an “arbitrary” distribution.

Wang and Japkowicz (2008) use boosting together with asymmetric datasets and produce a set of SVM-based weak learners that collectively perform significantly better, not only for the larger class, but also the smaller one. This is achieved by using a combination scheme of “boosting” - weighting samples from the rarer class as well as modifying the data distributions of the classes to take into account the number of observations in each target class.

Pham and Cham (2007) use a similarly formulated combination boosting scheme with asymmetric datasets and develop an online training algorithm applied to the problem of face-detection.

The final group classification by the combination of weak learners can be regarded as an ensemble classification. In the above examples, given that the sizes of each learning sub-set in the sequence of learning sub-sets are smaller than the original dataset, it is useful to re-sample from rarer classes in order to ensure each class is always represented in every learning sub-set as well as boosting through an appropriate weighting scheme. Freund and Schapire (1996) describe two related methods: AdaBoost.M1 and .M2, which evaluate the probability of misclassification of each observation (by an ensemble of classifiers) and arrange the sampling probabilities of observations to be higher for those sample classes most often misclassified.

In the case of regression problems, boosting can also be applied (Freund, 1995) in a generalisation that allows errors to be spread more evenly across the dataset of observations, than in the non-boosted case. This is achieved by

resampling according to a distribution that is a function of the previous weak learner model error for each sample.

An excellent review of these techniques and a comparative explanation is contained in Maclin and Opitz (2011).

2.7.4 Based on feature selection

The objective of feature selection ensemble generation is to improve the performance of the resulting regressor or classifier models either by eliminating irrelevant and/or redundant information from the used input feature set or by building diversity using representative yet different subsets of features for each ensemble member.

Cherkauer (1996), in a project to identify volcanoes on Venus, created ensembles of 32-ANNs using randomly selected different subsets of 119 available input features – and also varied ANN architecture.

2.7.4.1 Fuzzy rough feature selection and harmony search

This subsection highlights feature-selection approaches to ensemble creation that focus on evaluating a given feature subset as a whole instead of measuring on an individual feature basis. Harmony Search (HS) (Lee and Geem, 2005) is an optimisation algorithm, similar to genetic algorithms (GA) (Holland, 1975) that is not inherently a feature-selection technique. However, it has been adapted (Diao and Shen, 2012; Shen et al., 2012) to the task of feature selection and the production of model ensembles. They have also employed fuzzy rough feature selection techniques for this (Diao and Shen, 2011; Jensen and Shen, 2009).

GA relies on crossover and mutation to generate (hopefully) fitter offspring solutions, but these strategies may not be well-suited to all problem domains. HS uses the principle of finding a single or multi-objective global optimum by seeking "perfect harmony" between the values ("notes played") of decision variables ("players"). This is achieved through probabilistically selecting values from a previously tried harmony memory or from a random selection of untried values and then performing small perturbations ("improvisation") of them. This

is analogous to the function of differential mutation in GA. Typically the probability of selecting random untried values starts large and decreases during execution of the algorithm, rather like simulated annealing. Diao and Shen (2012) have adopted the technique to feature selection by allowing a differing number of "players" = decision variables = features in the search space. These are compared with standard hill-climbing, GA and Particle-Swarm Optimisation (PSO) (Coello et al., 2004; Kennedy and Eberhart, 1995).

Harmony Search Feature Selection (HSFS) (Diao and Shen, 2012) uses a 2-D multi-objective optimisation approach (feature subset size and subset performance evaluation) to search for an optimal subset of features using an adapted HS algorithm. In this, the "players" are no longer features, but feature selectors and the "harmonies" become combinations of features. The feature-selectors are allowed to select any feature or none. Where 2 or more "players" select the same feature, this is selected only once. The number of feature-selectors, N, needs to be set by the user. Intuitively, this might be expected to equal the total number of features, but in practice, improved results are frequently found to be obtained with a lower number. The method therefore starts with N large and reduces it iteratively. Diao and Shen combine HS with the use of fuzzy-rough sets (Beaubouef and Petry, 2012) to determine probabilistic membership of any feature to a given "reduct" (subset of features). They then evaluate performance of each reduct in the second-objective and so determine an optimal reduct, which will be located close to the knee-point of the Pareto-front of non-dominated solutions held in Harmony Memory at the completion of execution. However, from the paper it does not appear that this has been fully taken into account. Instead, it appears that only reduct size may be used to terminate execution. Nonetheless, the method allows possible feature pairs or groups that jointly form an informative feature subset to be found, which is important in many real datasets. The application of fuzzy-rough sets is equivalent to providing each feature with a weighting factor that determines its contribution to the class membership computation for a given reduct. In this regard, the method is similar to use of a single-layer ANN for each reduct. The methodology uses 10-fold leave-one-fold-out-cross-validation (NFCV) to test, so as to ensure that all samples in the dataset are evaluated as

part of an aggregated test dataset, similar to the method described in Chapter 4.

In another paper (Shen et al., 2012) the method is extended to construction of ensembles of ANNs. This is effectively a wrapper-based approach, so is covered in section 2.6.1.

2.7.4.2 Principal component analysis for ensemble construction

Principal Component Analysis (PCA) as described in section 2.6.2.2 is a broadly applicable approach to data dimension-reduction and feature selection.

For ensemble construction, the entire set of models can use the same reduct of principal components as input features. Alternatively, each ensemble member can use a different reduct. This may lead to improved diversity within the ensemble, but it is also possible that using low ranked principal components as input may not lead to significantly better model performance and may even degrade it, depending on the extent to which the selected components merely represent unstructured residuals (noise) in the data. Decision Tree ensembles are developed (Rodriguez et al., 2006) using PCA as a feature-selection technique.

2.7.4.3 Wrapper-based feature selection for ensembles

In (Shen et al., 2012), a matrix of methods are evaluated (feature selectors x classifier algorithms) to produce ensembles of classifiers in which each member may contain reducts of features different from each other (also different numbers of features). The feature selectors used include: fuzzy-rough feature selection (FRFS) (Jensen and Shen, 2009); correlation-based feature selection (CFS) (Hall, 1999); and probabilistic consistency-based feature selection (PCFS) (Dash and Liu, 2003). Furthermore, members implement a number of different classifier algorithms, such as decision tree based C4.5 algorithm or rule based Ripper algorithm (Witten and Frank, 2005), and vaguely quantified fuzzy-rough nearest neighbour (Jensen and Cornelis, 2008) to create truly diverse ensembles. The ensembles are tested against 11 of the real-valued UCI datasets (Bache and Lichman, 2013). A 10-fold stratified cross-validation approach is used to divide the training and test data; thus generating

ensembles of 10-members, similar to the approach described in Chapter 4 of this thesis. The stratification prior to division into folds ensures that each class is represented by an approximately equal number of samples, so as to attempt to avoid problems with bias and variance of the samples within each fold. However, this would also tend to have an adverse effect on the diversity within the ensemble, due to the potentially multiple repetitions of the same samples belonging to the most under-represented classes. These would then tend to appear in every fold, producing a normative effect on the statistical distributions of the folds. In summary, no feature-selector or classifier has emerged as a clear leader. However, the feature selectors are able to produce reducts considerably narrower than the original datasets without degradation of performance. Due to the complexity of this trial, no attempt is made to pre-optimise the parameters of each algorithm used; so this introduces a distinct weakness in its findings. The approach of using a diverse range of methods together to form ensembles has merit in terms of the likely robustness of the aggregate of classifiers. This approach is equally applicable to the use of ANNs as feature-selected classifier ensembles although they are not implemented in this paper.

2.7.5 Neural network ensembles

The approaches to ensemble generation covered in sections 2.7.1 to 2.7.4, although not all specifically tested on ANNs, are included because the principles are applicable to neural networks. In this sub-section, existing research specifically on ensembles of ANNs is examined.

More than twenty years ago, Hansen and Salamon (1990) established that ensembles of ANN classifiers could outperform the best individual classifier in an ensemble using a majority voting scheme to determine predicted class. They also showed that diversity in the ensemble derived from a number of sources, including randomised initial values of the network weight vectors and the training of different members on different subsets of the dataset. Improved performance of the ensemble is explained in terms of members discovering different subsidiary maxima ("traps") in the space of classification rate versus weight values; thus they are collectively able to generalise better than even the best-performing individual ensemble member. Statistical explanations for

improved performance of ensembles of ANN classifiers over their individual members is proposed in Dietterich (2000). In simple terms, this is due to a much reduced probability of misclassification for all members of the ensemble simultaneously.

EAs maintain populations of solutions and apply a process akin to natural selection on that population in order to generate fitter solutions in each generation. This naturally lends itself to generation of ensembles of ANNs, since each solution in the population is an ANN. For example a set of non-dominated (rank 1) solutions on completion of a multi-objective EA could form an ensemble of ANNs. Despite lack of consensus on diversity measures/metrics to use, it is generally agreed that it is good to maintain diversity within an ensemble of models, since this has been shown to improve robustness, reliability and performance of the ensemble as a whole (Cunningham and Carney, 2000; Kuncheva and Whitaker, 2003; Zhou and Li, 2010). Depending on the algorithm used, diversity within the ensemble can be arranged against a number of criteria; for example use of crowding-distance in NSGA-II (Deb et al., 2002b) combined with definition of objectives in the fitness function that measure factors such as architectural-complexity or weight-regularization; or use of correlation penalty (Liu and Yao, 1999a, 1999b) in the fitness function (evaluated across the whole ensemble together during simultaneous training of all members).

Liu and Yao first demonstrate a method they designate CELS (Cooperative Ensemble Learning System) for regression ANN models; then apply it to classifiers as Negative Correlation Learning (NCL) (Chen and Yao, 2010; Liu et al., 2000; Liu and Yao, 1999a; Wang et al., 2010). The latter method takes into account that classification rate of the ensemble as a whole is a function of both the variance within each model and the covariances between them. Both CELS and NCL attempt to correlate negatively the errors made between combinations of members in the ensemble. An essential feature of both methods is the simultaneous training of all members of the ensemble, which allows the unsupervised correlation penalty term to be applied equally to all members of the ensemble, instead of progressively to later-trained ensemble members. Effectively, NCL attempts to ensure that different ensemble

members make different classification errors; thus improving overall performance. Various algorithms have been proposed and the reader is referred to (Chen and Yao, 2010; Liu et al., 2000; Liu and Yao, 1999a; Wang et al., 2010).

Unlike the methodology described in this thesis, none of these use analysis of weights and/or neural pathway strengths within the ANNs produced. However, similar to the above, in the method proposed here an entire ensemble is built and then the relevant features are selected, based on the parameters from the whole ensemble.

2.7.5.1 Application of NFCV to ANN Ensemble Generation

In several of the existing methods described in the previous section, the same number of models as data folds is separately trained on slightly different data dependent upon the statistics of the excluded fold in each case. It has been established that this set of models can be used as an ensemble (Shen et al., 2012). In section 2.5.1 this approach is defined as an N-Fold-Cross-Validation (NFCV) ensemble. Figure 2.9 illustrates a data-division schema used to generate such an ensemble of 7 ANN models. This schema is applicable to any number (>=3) of data folds and models and is widely used for ensemble generation. It is worth noting in Figure 2.9 that an additional (eighth) data fold (shaded light-blue) is retained for testing the entire ensemble of models following completion of training and test of all individual ensemble members. This is important for example where an aggregate prediction from the entire ensemble is to be evaluated or where comparisons between ensemble members' responses to the same set of observations are to be made.

Even if all ANNs in the ensemble are initialised to the same state (values of weights and biases) prior to training, their states will differ following training due to the slightly differing statistics of the samples within each training subset. These differences can be exploited in order to determine relevance of each input signal to the models, as will be demonstrated. Nonetheless, identical initialisation of ensemble members is not a requirement of the approach. It is robust against a number of architectural factors, such as initialisation and number of units in the hidden layer.

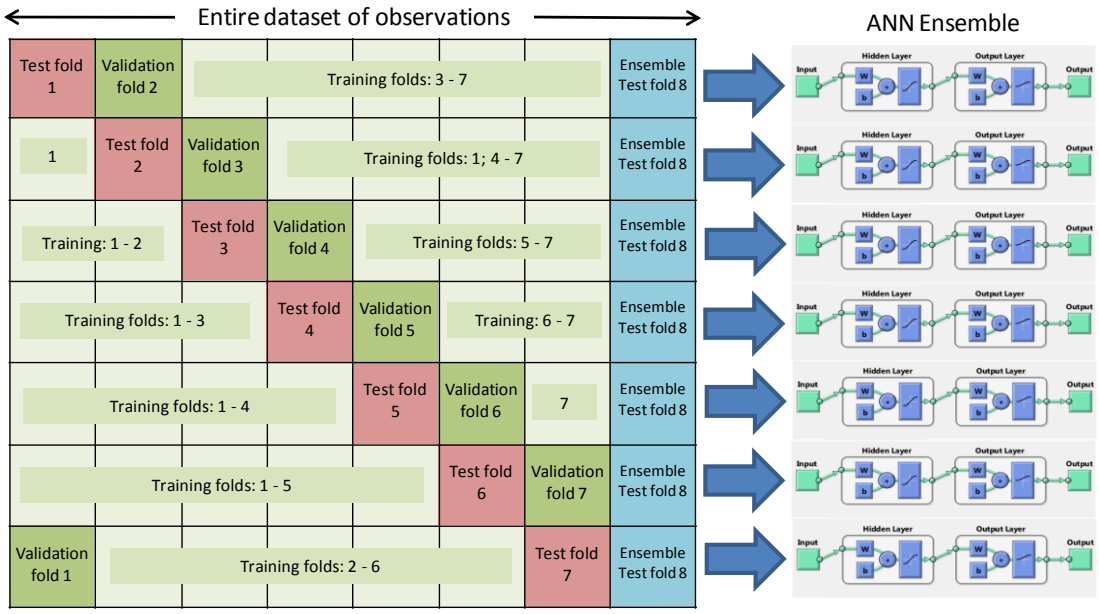


Figure 2.9. NFCV Model Ensemble Generation Schema

2.7.6 Other ensemble generation approaches

2.7.6.1 Generalised likelihood uncertainty estimation (GLUE)

The Generalised Likelihood Uncertainty Estimation (GLUE) methodology (Beven, 2006; Beven and Freer, 2001) allows for the “equifinality” concept; namely that there may be several models that perform acceptably in reproducing the observed behaviour of a system. Equifinality rejects the concept of a single optimal model. GLUE also allows for contributions of models within an ensemble to be weighted according to their likelihood measure to estimate prediction quantiles. The weighting for each model is time variant. GLUE uses a Bayesian framework for calculation of posterior probabilities, based on estimation of prior probabilities from a moving time-window, *T*, which is a subset of the entire observation dataset. A (usually very large) ensemble of models is created by Monte-Carlo Markov-Chain (MCMC) simulation (Ahmed, 2008; Hastings, 1970) or Latin Hypercube Sampling (McKay et al., 1979) to select parameter values for each model at each timestep. Posterior probabilities are then computed; models are then weighted as above and contribute accordingly to the overall prediction of the ensemble. For each model (sample of parameters) the posterior from the previous timestep is used as the prior for the next timestep. This data assimilation approach allows for non-stationarity, heteroscedasticity and other statistical measures of the observed time-series to be time-variant and so is flexible in this regard but, due to use of a moving time-

window sample of the dataset at each timestep, *t*, it appears to lack 'slow' model learning capability from long timescale information in the observed dataset. Naturally, the time-window could extend with each new timestep, such that all observations were included in the window, but this would mean that run-times would continue to extend as the number of observations continued to increase. This would therefore be unfeasible for real-time prediction. Questions are also raised regarding the validity of "informal likelihood" measures used in GLUE in a paper by Smith et. al. (2008).

A study on the Nigorikawa urban river basin in Kofu City, Honshu, Japan (Hapsari et al., 2011) uses a GLUE ensemble approach to prediction of flash flooding. The ensemble is generated by perturbing the initial condition (x-y direction) of the radar echo advection model to produce an ensemble of rainfall predictions. The 9-parameters of the advection model (allowing for a growth-decay term) are calibrated in real-time, using the Bayesian framework of the GLUE methodology. Use of the ensemble allows flood-risk and damage maps of the main area of the city to be generated, using a physically-based hydrological model of surface and sub-surface flows for each rainfall ensemble member. Predictions of up to 6-hours have been attempted, but performance is found to be satisfactory up to 3-hours ahead; the assumption of linearity for rainfall extrapolation is considered to break down beyond 3-hours.

2.7.6.2 Ensemble transformation and adaptive observations

The Ensemble Transformation and Adaptive Observations (Bishop and Toth, 1999) is a meteorological data assimilation technique used in Numerical Weather Prediction (NWP). It aims to solve the predictive challenge arising from the fact that regions of the atmosphere and oceans will vary in their influence on the weather everywhere from day to day. Effectively it is an ensemble approach using feature selection (adaptively introducing new features in the form of meteorological observations at new locations and/or removing them again) based on identification of regions of high influence for a period of time. The technique is based on building and maintaining error covariance matrices based on inclusion/exclusion of observed features. As in standard Ensemble Prediction Systems (EPS) data assimilation approaches (Cloke and Pappenberger, 2009), perturbations of an optimal analysis in terms of

trajectories of development of the state of the system (e.g. Lyapunov vectors (Toth and Kalnay, 1993)) are used to create members of an ensemble of forecasts. Although pure Monte Carlo perturbations were originally used, these were found to be relatively ineffective. Instead, ETAO uses a method of creating a small random perturbation; propagating both the optimal analysis forward by the breeding cycle period (say 6-hours) and then recalibrating the perturbed forecast to have the same covariance as the original perturbation. Since these perturbations are reinserted into the analysis at each breeding cycle, the most rapidly developing modes of the atmosphere are represented in the set of ensemble members. In summary, ensemble members are created, both by (normalised) random perturbations on a mean ("optimal") analysis and by dynamic selection of observational features (here in different spatial regions of the atmosphere). See Figure 2.10.

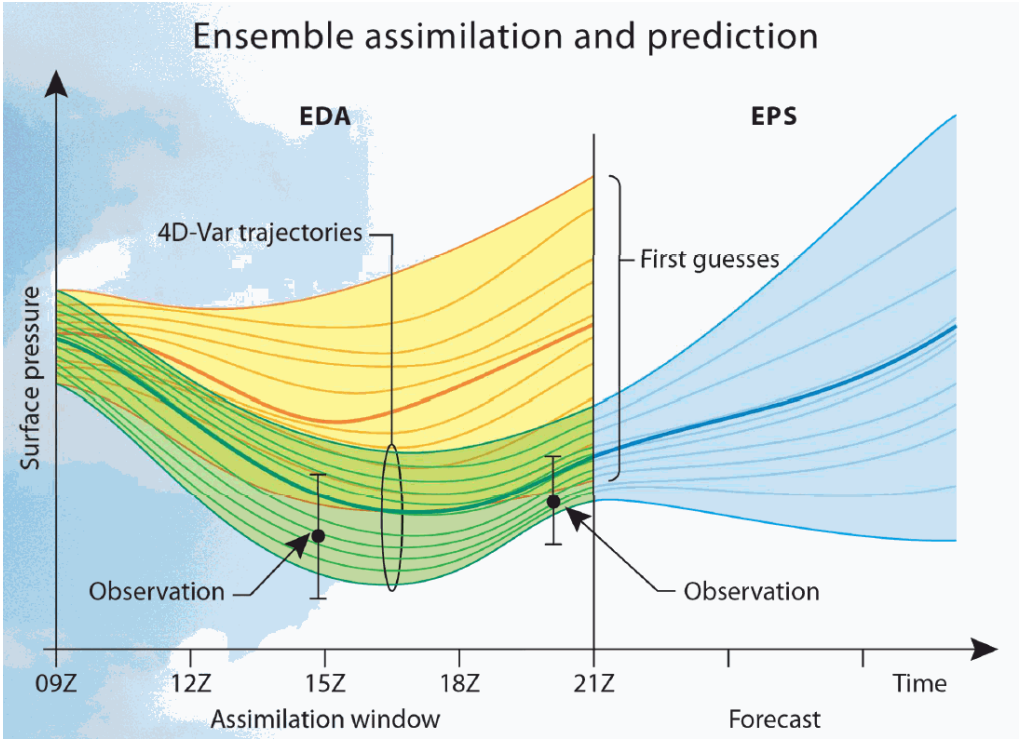


Figure 2.10. Ensemble Assimilation and Prediction (Magnusson, 2012)

2.8 Opening up the black box

ANNs are examples of Data-Driven Models (DDMs) and implement a non-linear multi-dimensional transfer function between a set of input signals and a

set of resultant outputs[9]. This is often referred to as a "black box" (Sjöberg et al., 1995) with the implication that the transfer function itself is the only known property of the model; the means by which the transfer function is achieved is implicitly unknown. This could arguably explain the relatively slow uptake of DDM techniques in live systems within hydrology and the water industry to date, despite an enormous amount of research having been successfully conducted. Improved tools to open up and analyse the operation of neural network black boxes are thus sought.

2.8.1 Grey box techniques

A large section of literature discusses “grey-box” ANN models (Acuña et al., 1999; Chen et al., 2011; Millie et al., 2012; Thordarson and Madsen, 2012). These combine some prior knowledge of physical processes and/or parameters with available observation data. So doing, they aim to achieve “the best of both worlds” between purely physically-based “white-box” models and fully data-driven black-box models. Figure 2.11 illustrates. In general they distinguish between observation noise and process noise and aim to provide sufficient model parameterisation so that no obvious structure remains in the process noise. That is to say that they are at that point fully-calibrated models. Acuña et al. (1999) describe two approaches that either explore and act on models’ own parameters (“direct”), or they use ANNs to calibrate the parameters of other models (“indirect”) that may or may not be themselves based on ANNs. In this way the grey-box approach does externalise and explore models’ structure and parameterisation.

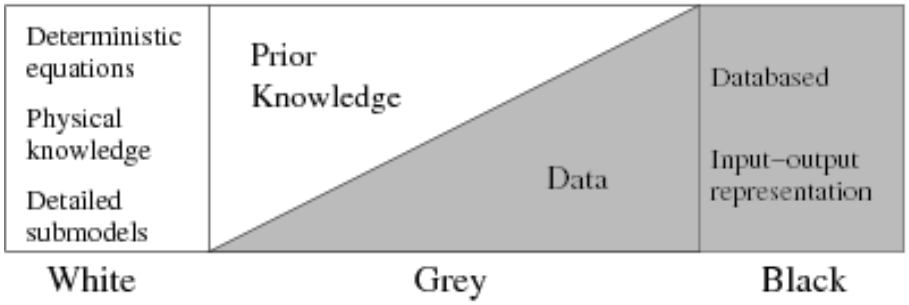


Figure 2.11. White, grey and black box models (Thordarson and Madsen, 2012)

[9] *In the case of feedforward Multi Layer Perceptrons (MLPs) these transfer functions are nevertheless fully deterministic.*

2.8.1.1 Illuminating the "black box": randomization approach

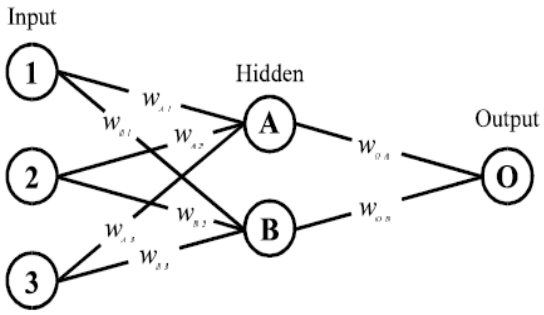
One grey-box approach is worthy of detailed description (Olden and Jackson, 2002) since it, to a significant extent, overlaps with the approach described here; yet it also differs in a number of key respects. It performs analysis of ANN weights using both Garson's algorithm (Garson, 1991) and an adaptation of it to perform both pruning of internal connections and assessment of relevance of input features, which enables their selection. The case study involves producing an ecological model to predict number of fish species present (1 to 23) in 284 freshwater lakes in Ontario, Canada as a function of (up to) eight habitat-related predictor variables. Figure 2.13 provides an example of a useful method of visualisation of such a neural network, Network Interpretation Diagrams (NID).

Figure 2.12 provides details of Garson's algorithm, used to evaluate the relative importance of each input as a predictor of the output. Interestingly Garson uses absolute magnitude of weights rather than taking their sign into account, which, as Olden and Jackson correctly point out, loses vital information about the interactions between the effects of hidden neurons. For example hidden neurons may work so as to cancel the effect of each other out; thus reducing the overall effect a given input has on the output. Garson's algorithm misses this. Olden and Jackson therefore adapt the method and calculate effects taking signs of weights into account. Their method for computing input-hidden-output weight influences parallels the approach described in Chapter 4 of this thesis. However, it is not clear that the previous authors realise that their computation could be effected more simply with a matrix multiplication.

A key innovation of Olden and Jackson is the technique of randomly permuting the target output samples (i.e. disordering them with respect to the input samples), then retraining the network starting from a fixed initial value of weights that produced a well-performing model with the true dataset. This is repeated in their experiment 999 times, in a similar fashion to a Monte-Carlo simulation, so as to build up a randomised probability distribution of input-hidden-output influences for each input. This is then compared with the true (best) model to see if the influence is outside of the 90% or 95% percentile range of the randomised distribution. If it is, then the input is statistically

significant and can be retained in the pruned model. Each pair of connections via each hidden unit can also be treated in this way and a decision made about pruning out or retaining the connection pair. Figure 2.13 illustrates a pruned network after connections falling inside the 95^{th} percentile of the randomised weight distribution have been removed. It is also worth noting that 3 of the 8 inputs are thus left unconnected and could be removed.

Olden and Jackson's approach is ingenious and gives a reasonable statistical basis for decision making about network architecture and input feature selection. The determination of the 95 percentile points using a t-test would of course assume the Gaussian nature of the probability distributions, which may not necessarily be the case. This difficulty could be overcome by counting cases starting from the extremes until a 5% point is found. Perhaps the biggest drawback of the method would be its very high computational cost, since it requires 1000+ ANNs to be created, trained and their input-hidden-output weight influences calculated in addition to performing 999 fully randomised permutations of the dataset. A further discussion of this methodology in comparison with the method proposed in this thesis is included in Chapter 4.



1. Matrix containing input-hidden-output neuron connection weights

	Hidden A	Hidden B
Input 1	$w_{A1} = -2.61$	$w_{B1} = -1.23$
Input 2	$w_{A2} = 0.13$	$w_{B2} = -0.91$
Input 3	$w_{A3} = -0.69$	$w_{B3} = -2.09$
Output	$w_{OA} = 1.11$	$w_{OB} = 0.39$

2. Contribution of each input neuron to the output via each hidden neuron calculated as the product of the input-hidden connection and the hidden-output connection:
e.g., $c_{A1} = w_{A1} \times w_{OA} = -2.61 \times 1.11 = -2.90$

	Hidden A	Hidden B
Input 1	$c_{A1} = -2.90$	$c_{B1} = -0.48$
Input 2	$c_{A2} = 0.14$	$c_{B2} = -0.35$
Input 3	$c_{A3} = -0.77$	$c_{B3} = -0.82$

3. Relative contribution of each input neuron to the outgoing signal of each hidden neuron: e.g., $r_{A1} = |c_{A1}|/(|c_{A1}| + |c_{A2}| + |c_{A3}|) = 2.90 / (2.90 + 0.14 + 0.77) = 0.76$;
and sum of input neuron contributions: e.g., $S_1 = r_{A1} + r_{B1} = 0.76 + 0.29 = 1.05$

	Hidden A	Hidden B	Sum
Input 1	$r_{A1} = 0.76$	$r_{B1} = 0.29$	$S_1 = 1.05$
Input 2	$r_{A2} = 0.04$	$r_{B2} = 0.21$	$S_2 = 0.25$
Input 3	$r_{A3} = 0.20$	$r_{B3} = 0.50$	$S_3 = 0.70$

4. Relative importance of each input variable:
e.g., $RI_1 = S_1 / (S_1 + S_2 + S_3) \times 100 = 1.05 / (1.05 + 0.25 + 0.70) \times 100 = 52.5\ \%$

	Relative importance
Input 1	52.5 %
Input 2	12.5 %
Input 3	35.0 %

Box 1. Garson's algorithm for partitioning and quantifying neural network connection weights. Sample calculations shown for three input neurons (1, 2 and 3), two hidden neurons (A and B), and one output neuron (○).

Figure 2.12. Garson's Algorithm reproduced from Olden and Jackson (2002)[10]

[10] *Reprinted from Ecological Modelling Vol 154, Authors: Julian D. Olden and Donald A. Jackson Illuminating the "black box": a randomization approach for understanding variable contributions in artificial neural networks, Pages No. 135–150, Copyright (2002), with permission from Elsevier.*

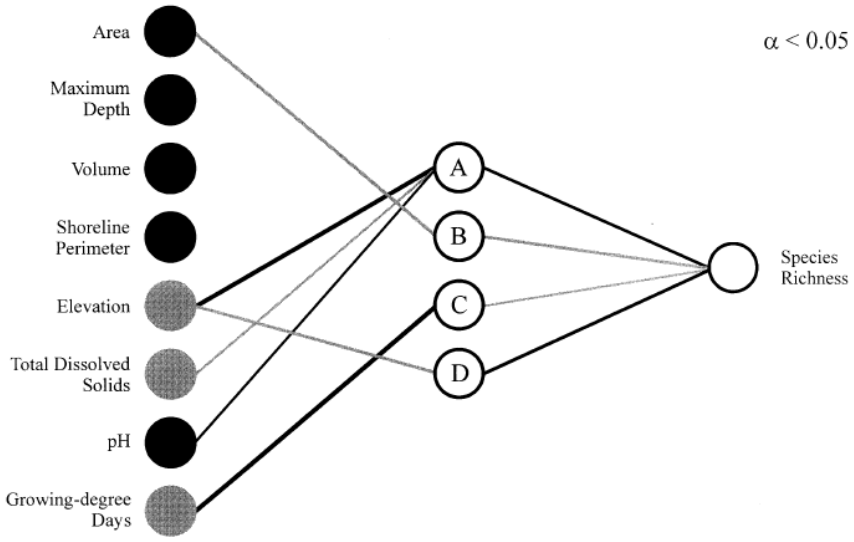


Figure 2.13. NID Pruned using 95% statistical significance (Olden and Jackson, 2002)

2.8.2 Visualisation

As a result of the research into ANN weight and bias values in this thesis, visualisation techniques for viewing the internal operation of 2-layer feedforward neural networks during training are also developed. This reveals the structure of “morphemes” and “sememes” (Hinton, 1984; Hinton et al., 1993) within a 2-dimensional neural pathway strength space and its breakdown into three 2-dimensional subspaces organised by output neuron, hidden neuron or input signal. Neural network visualisation techniques are also therefore reviewed in this section.

Perhaps the earliest graphical method developed for visualisation of the weight and bias structure in an ANN is the Hinton diagram (Rumelhart and McClelland, 1986b). These represent the absolute values of each weight or bias by the size of a square within the figure, whilst positive values are represented originally by white squares (green in the illustration) and negative values by black squares (red in the illustration) on a grey background. In the original text, separate sub-plots are produced for each hidden unit in a network. In the example Hinton diagram of Figure 2.14 (standard format output from the MATLAB (Mathworks, 2012) 'plotwb' function) positive values are represented by green and negative by red squares. The illustrated network has 5 inputs, 10 hidden neurons and 10 outputs.

Figure 2.14(a) shows the state of the ANN following initialisation of the weights and biases to random values, whilst (b) illustrates its state on completion of training. From this, it is possible to observe certain patterns, such as that for input 3 (third column in top-left rectangle) there is a trend from positive to negative for the use made of it by hidden units 1 to 10 (rows 1 to 10 of that column); and that outputs (rows in bottom-right square) generally make stronger use of the output from hidden unit 1 (first column in bottom-right square) than (say) hidden unit 6 (sixth column). It is true to say that Hinton diagrams visually represent weights rather than neural pathways, however.

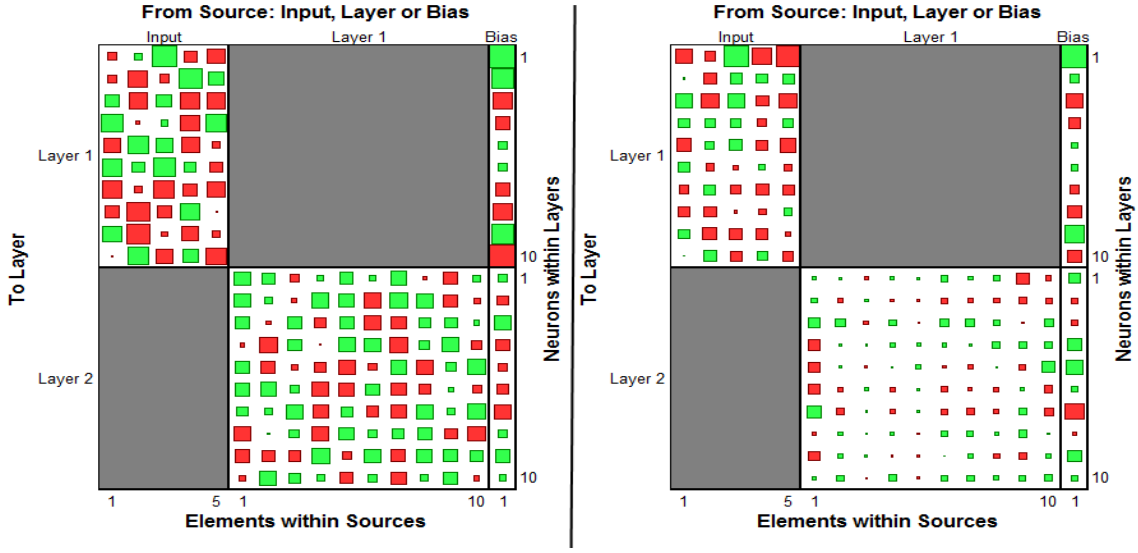


Figure 2.14. (a) Hinton diagram - before training; (b) Hinton diagram - after training

Neural Interpretation Diagrams (NIDs) are used in relation to the approach described in section 2.8.1 using Garson's algorithm to effect network connection pruning and input feature selection (Olden and Jackson, 2002).

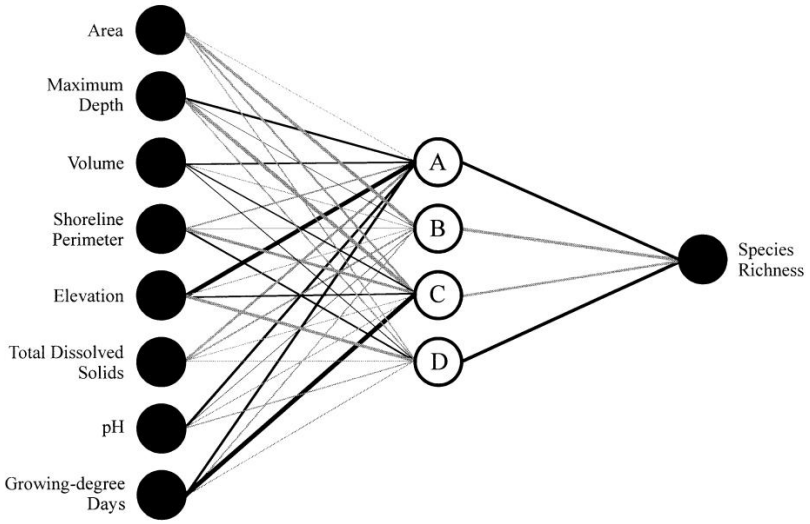


Figure 2.15. NID for neural network modelling fish species richness as a function of eight habitat variables (Olden and Jackson, 2002) [11]

These represent positive values of weights as black lines and negative values of weights as grey lines, whilst the magnitude of value of weight is represented by the line thickness. NIDs are effectively directed graphs – applied in their paper to layered feedforward networks only – that give a good qualitative representation of the connectivity within an ANN. However, they suffer from poor quantitative information as well as the psychological difficulty that black lines and grey lines of the same thickness may not be perceived as such by human beings; with more weight being attributed to the black lines.

A number of papers based on variations of the ARTMAP neural network (for Adaptive Resonance Theory Mapper) ((Amis and Carpenter, 2010; Carpenter et al., 2005, 1998, 1997a, 1997b; Liu et al., 2001) describe a system for classifying vegetation types from satellite imagery. Other ARTMAP applications are also described in the wider literature. These use 2D graphical representations of weight spaces occupied by the top-down and bottom-up classification weight vectors, which are a key element of the ARTMAP approach. Figure 2.16 illustrates an example of ARTMAP box plots using MODIS satellite Normalized Difference Vegetation Index (NDVI) spectral input vector weights for August in the x-axis and January in the y-axis. Each plot relates to a different output classification of vegetation and each rectangle

[11] *Reprinted from Ecological Modelling Vol 154, Authors: Julian D. Olden and Donald A. Jackson Illuminating the "black box": a randomization approach for understanding variable contributions in artificial neural networks, Pages No. 135–150, Copyright (2002), with permission from Elsevier.*

circumscribes the area into which each given NVDI pair has to fall in order to belong to the target class.

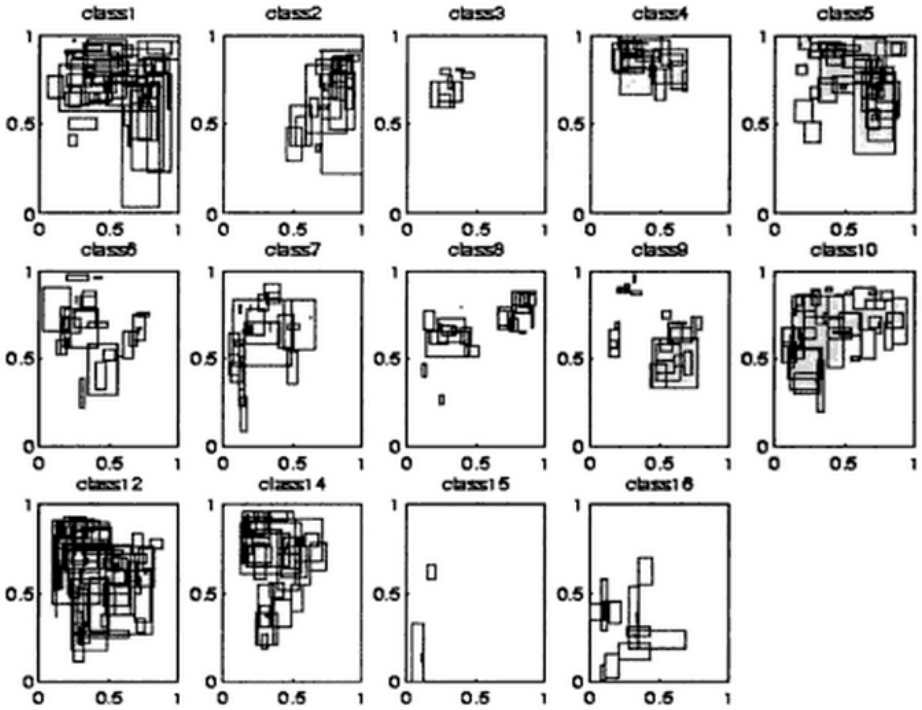


Figure 2.16. ARTMAP Box Plots of MODIS Classes [from (Liu et al., 2001)]

Minimal ART neural networks consist of 2-layers of neurons but connections are bidirectional (feedforward and feedback) between the layers, allowing a resonance phenomenon to occur identifying (typically) a single output class. The ARTMAP box plots can be used to visualise the upper and lower bound weights defining the classification regions both during and after training.

Bullmore and Sporns (2009) describe a graph theoretical approach to model and represent structural and functional pathways in brains. Figure 2.17 shows directed graphs of both structural and functional pathways in a brain. These were produced using correlation techniques including spectral coherence or Granger causality (Granger, 1969) between magnetoencephalography sensors (functional), or the connection probability between two regions of an individual diffusion tensor imaging data set (structural). In the functional graph, nodes separated by >75mm are shown in blue; whilst those of length <75mm are shown in red (Achard et al., 2006). Although neural pathway strengths are not represented directly in these diagrams, the colour-coding of neural pathways based on their length is employed, so these are included here as an illustration of an alternative visualisation of neural pathways in a 2D colour plot.

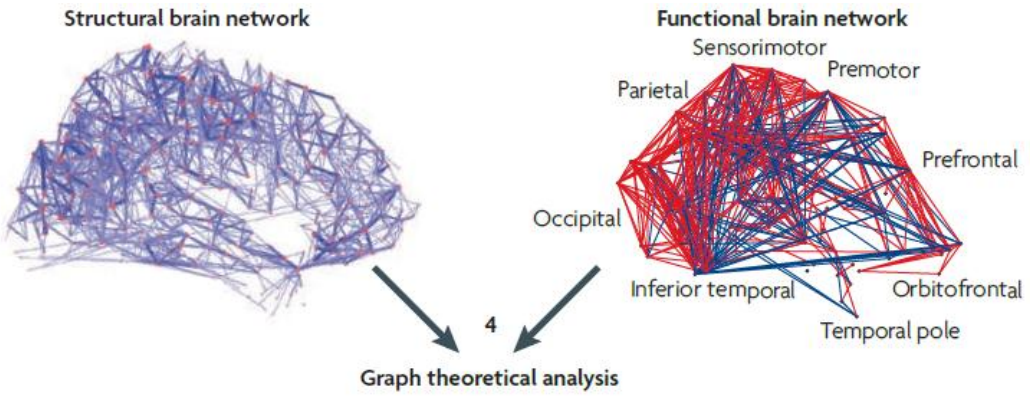


Figure 2.17. Structural and functional neural pathway graphs from Bullmore and Sporns (2009)

However, these graphs use a thresholding approach; correlation values below a certain threshold are not represented in the graph. Similarly the length classification threshold of 75mm has been used to determine colour used. Relative connection strength information is not represented in the directed graphs.

2.9 Applications

In order to demonstrate the applicability of ANNs to real-world problems in hydrology and the environment, two case study chapters are provided. The first of these covers fast ANN-based predictive surrogate models for urban flooding (Chapter 3). It describes the early research that led to the discovery of the machine learning techniques described in chapter 4. The second case study chapter covers bathing water quality prediction (Chapter 5). Two subsections are therefore provided in this review to cover the extensive research to date in these areas.

2.9.1 Urban flooding, sewerage and related applications

Recent years have seen considerable interest in the hydrological applications of ANNs, since, if sufficient observation data is available, they can obviate the need to build and calibrate physically-based models, for example based on hydrodynamic equations.

Data-Driven-Modelling (DDM) is a generic term for this approach to modelling relationships between input and output data, as opposed to

conventional modelling involving use of dynamic equations describing the underlying physics of the modelled system.

Once trained, ANNs also run significantly faster than hydrodynamic models, potentially allowing real-time early-warning systems to be created and run on the most humble of today's PC's or even with client apps on mobile devices. Moreover, they can be readily adapted either to predict continuous quantities or to classify. Flooding early-warning systems, for example, could predict flood depths or volumes of spill, or classifications of flood severity. Researchers have studied ANN application to pluvial, fluvial and urban drainage flooding. This has been so extensive that a comprehensive literature review is not attempted here. Instead a broad cross-section of the most relevant examples are selected and presented here.

Fernando successfully models and predicts outflows from a single sewer overflow (CSO) in the drainage system of Auckland, NZ, with a history of problematic spill events (Fernando, 2005). In this study, 6 antecedent overflow rates (generated by traditional simulator from 100-years historic rainfall data) and 12 antecedent rainfall data (from the nearest rain gauge) in a moving time-window regime are used to produce a prediction of sewer overflow rate of up to 9.5 hours lead time. However, the study shows an approximate Time of Concentration (ToC) for the CSO of between 2 and 2.5 hours. Predictive performance beyond 2.5 hours is seen to decrease significantly. A 3-layer, feedforward, multi-layer perceptron (MLP) is configured with 18 input nodes, 9 hidden neurons and 1 output node. The antecedent flow rates are found to be vital and that a modified system, purely based on rainfall was not very accurate at predicting overflow. The experiment no doubt benefitted from the lags inherent in the sewer network between the points of rainwater ingress and the CSO – increasing the limit of prediction advance achievable. The approach used in chapter 3 exhibits significant similarities to Fernando’s study. However, there are two key differences:

1. The need for use of antecedent flows as model inputs is problematic, when considering a live real-time EWS. The implication is that either these would need to be provided by a telemetered gauge or they would also need to be computed using a hydro-dynamic simulator on an ongoing basis. In the first

case, extra capital and maintenance costs would be involved for every CSO to be monitored. In the second case it is difficult to see what significant computational advantage the ANN model would provide over simply using the simulator also for the prediction, given that it would have to be run anyway. The models in chapter 3 work principally with rainfall input only; so avoid this issue.

2. A single CSO output is modelled in this study; whereas the approach used in chapter 3 models multiple sewer nodes with a single multi-output ANN.

In a Danish case study (Thorndahl et al., 2009), a data-driven model (DDM) is created to predict water levels at multiple nodes in a sewer based on rain radar images. The approach uses two stages:

1. A rainfall advection model based on the CO-TREC algorithm (Mecklenburg et al., 2000)
2. A sewer flood DDM based on the WaterAspects (Grum et al., 2004) modelling tool.

Whilst this is a first of its kind, in this study the WaterAspects model requires calibrating by hand by the process of adding more elements to the sewer model until the output matches the target hydrograph with sufficient accuracy; presumably a fairly effort-intensive process.

Chiang et al. (2010) use recurrent neural networks (RNNs) in a study of the sewerage system in Taipei, Taiwan to predict water levels at an ungauged and a gauged manhole, 5, 10, 15 and 20-minutes ahead. Target signals for the ungauged site are generated using the SWMM simulator. The use of RNNs provides a mechanism for the ANN to have memory of previous timesteps in the internal and output states of the ANN, which are fed back to inputs of neurons on each layer. This is used as an alternative to the approach of using lagged inputs in a moving time window.

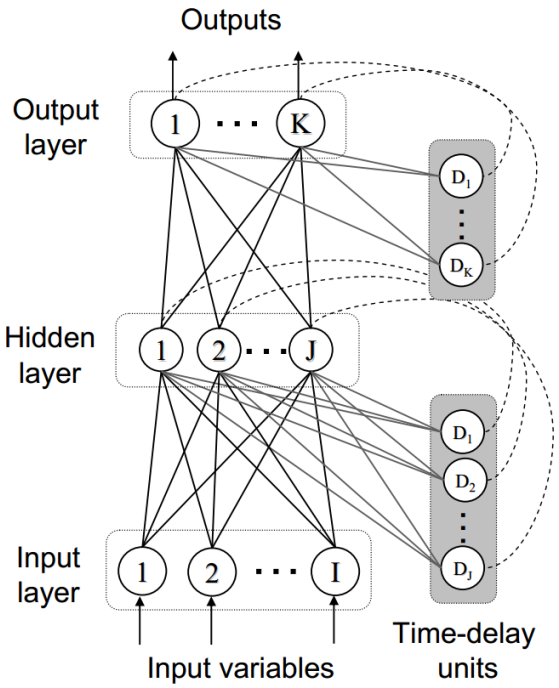


Figure 2.18. Recurrent Neural Network used in Taipei study (Chiang et al., 2010)

The eight RNNs modelled in the study each have a single output and model one of the four timing advances at either the gauged or the ungauged site. Results are very good, showing a very high level of accuracy with both R^2 correlation and Nash Sutcliffe Efficiency Coefficient (NSEC) scores above 0.97 for all four prediction advances at both sites.

In summary, the timescales for prediction of up to 20-minutes are modest and again, single sewer nodes are modelled by each neural network, rather than a multiple node approach in a single model.

Bruen and Yang (2006) compare the use of ANNs with linear auto-regressive (AR) and auto-regressive moving average (ARMA) models to apply corrections to the output of the HydroWorks (predecessor of InfoWorks) hydraulic model in the forecasting of flooding in an urban drainage network. Figure 2.19 illustrates how each of the above 3 types of black-box models is applied to the difference between HydroWorks output and the observed values from the sewer network to provide an additive correction to the estimate. The aim is to minimise the residuals to the level where they are completely uncorrelated with the observations or estimate.

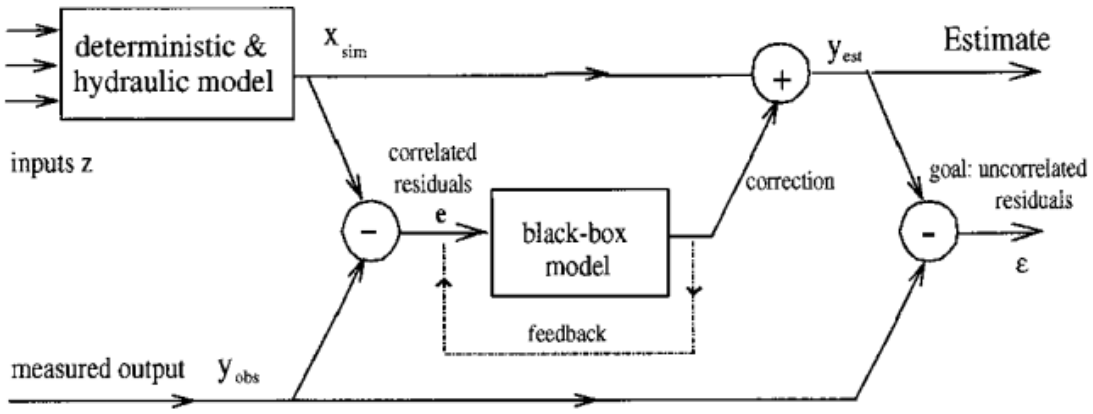


Figure 2.19. Using Black-box models to correct hydraulic simulator output (Bruen and Yang, 2006)

The case study is for a small, steep urban catchment in western Dublin, Republic of Ireland and the modelled location is a single node at the catchment outlet, for which gauged observations are available. The mean lag-time for this node is 39.4-minutes. All black-box models perform well at lead-times up to 8-minutes but the ANN R^2 efficiency deteriorates most rapidly at longer lead times, with the ARMA model giving the best performance at the advances of up to 30-minutes trialled.

2.9.2 Fluvial flooding, rainfall-runoff and related applications

Campolo (2003) in Italy uses ANNs to model flow rates and hence flooding of River Arno at Florence and successfully predicts these up to 6 hours in advance, which is an operationally useful warning period. This uses a novel approach to timeslicing sampled real data from sensors further upstream from the target site in Florence. Different resolutions of sampling rate are used depending on the proximity of data to the present moment. The optimal configuration employs 57 input-nodes, 30 hidden-nodes and 6 output-nodes. The time-slicing regime is illustrated in Figure 2.20.

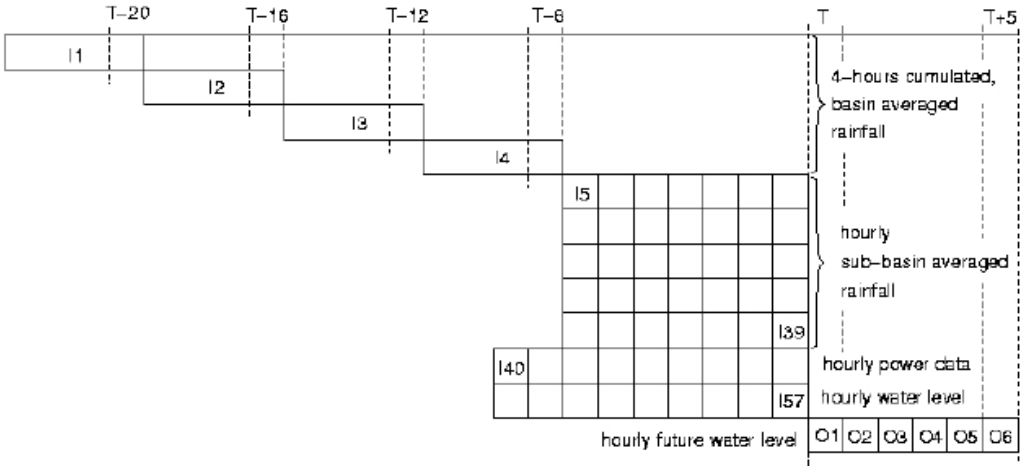


Figure 2.20. Input data used for water-level prediction up to 6 h ahead (Campolo, 2003)

An advantage of fluvial flood prediction over urban flood prediction is the significantly longer transport times (Times of Concentration – ToCs) of the order of 6-hours within the River Arno catchment, rather than tens of minutes in the case of urban drainage networks. The ANN model is aided in achieving 6-hour prediction time by these transport times within the catchment.

Significant similarities, however, between this study and the research in this thesis are:

1. The use of multiple ANN outputs. Here they are used to predict at more than one timestep ahead: time T to T+5 hours in this case. (O1 to O6 in Figure 2.20) rather than at different locations in the catchment.
2. The use of spatially varying rainfall as ANN inputs – here averaged over quite large sub-catchment areas.
3. A moving lagged-input time window – in this case with selection of used lags.

Dmitri Solomatine has published several papers in which DDM techniques are described that evolve ANNs to model fluvial flows (Solomatine, 2007a, 2007b, 2008). Notably, the hybrid “modular model” approach: splitting and recombination techniques to allow separate ANNs to model for example baseflow and peak flow are employed. This is illustrated in Figure 2.21 and Figure 2.22. Each model is referred to as a "local" or "expert" model.

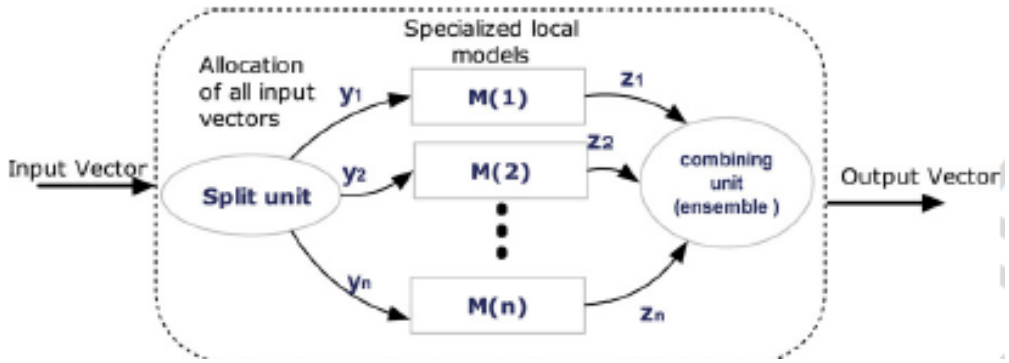


Figure 2.21. Modularisation Approach to Hydrological Modelling using ANNs (Solomatine, 2007b)

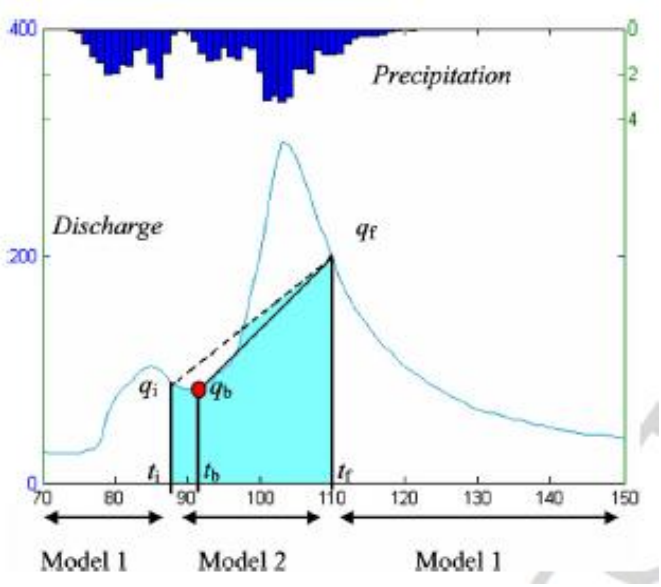


Figure 2.22. Modelling of different phases of the hydrograph using modular models (Solomatine, 2007b)

Different phases of the hydrograph can be modelled using different DDMs. In his papers, Solomatine and his team use both linear Model Trees and ANNs for the modular models. Combining of the modular models is achieved either by means of logic switches, or by use of fuzzy-logic switches in, for example, an adaptive neuro-fuzzy inference system (ANFIS) – a combination of ANNs and fuzzy logic. For example an ANFIS model is used for reconstructing missing flow data from monitored gauging station time-series using rainfall data and flow data from neighbouring stations (Dastorani et al., 2010). Both pure ANN and ANFIS show better results than conventional linear correlation-based approaches. The idea of using several models in combination means that each

model can specialise on the particular scenario it covers. If they are combined correctly, the overall result is demonstrably better than for single models.

Solomatine's group at UNESCO-IHE has also experimented with use of GAs in the feedback loop of ANNs and model trees used as flow models, to optimise network weights (Solomatine, 2008). See Figure 2.23. In this example model trees are evolved using the GA over a number of training steps, whilst optimising fitness function based on correlation with training flow-rate data.

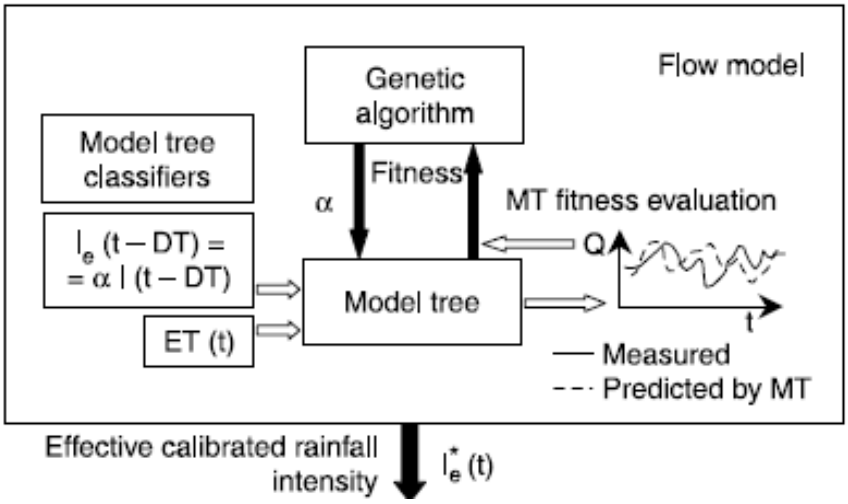


Figure 2.23. Hybrid-Model-Tree GA Scheme for Flow-Modelling (Solomatine, 2008)

Goswami and O'Connor (2007) use a multi-model approach to prediction of flow rates up to 6-days ahead in the Brosna river in Ireland and a river in France. Two scenarios are tried:

1. Using only antecedent flow data as input
2. Using antecedent flow and rainfall data as input

No QPF (precipitation forecast) data were available. ANN models are compared with linear auto-regressive (AR) and parametric simple linear (PSL) models in both scenarios. In general the ANN models perform better than their linear counterparts, but the best results are obtained by combining all the models to form a composite prediction. The models including the rainfall data as inputs also perform better than their flow-only counterparts.

Abrahart et al. (2007) develop ANN models to predict flow rates in the river Ouse (N. Yorks, UK) using rainfall and gauging station flows as inputs. They reflect that timing errors in such models have been reported in the literature, in which forecast flows are advanced in time from the target. They develop and apply an additional performance metric for the training of the ANNs, which penalises timing errors by a factor of 500. The results are evaluated for predictions 6-hours and 24-hours in advance and they find that low-flow prediction accuracy is improved for the 6-hour predictions, whereas (more importantly) high flow rate predictions are improved for the +24-hour prediction, when the timing error correction is applied.

Corani and Guariso (2005) optimise ANN fluvial flood models by using a pruning algorithm, Optimal Brain Surgeon (OBS) (Hassibi and Stork, 1993) to remove connections from a fully-connected 1HL feedforward network. Effectively this is a method for removing (in the limiting case) non-salient inputs from the network altogether. In their study an average of 30-40% of inputs are found to be removed by this method. The result is a more parsimonious ANN that exhibits improved performance in relation to fully-connected networks. Again, only a single output predicts water level just one timestep ahead. Training involves multiple re-runs, one for every new pruning of the network, so is computationally demanding. Figure 2.24 shows the reduction in RMSE validation error as the OBS pruning algorithm reduces the number of weights and biases in the network (by removing connections). The runs start at the top right and progress towards the left. The finally selected architecture is the one with the minimum value of summed training and validation errors.

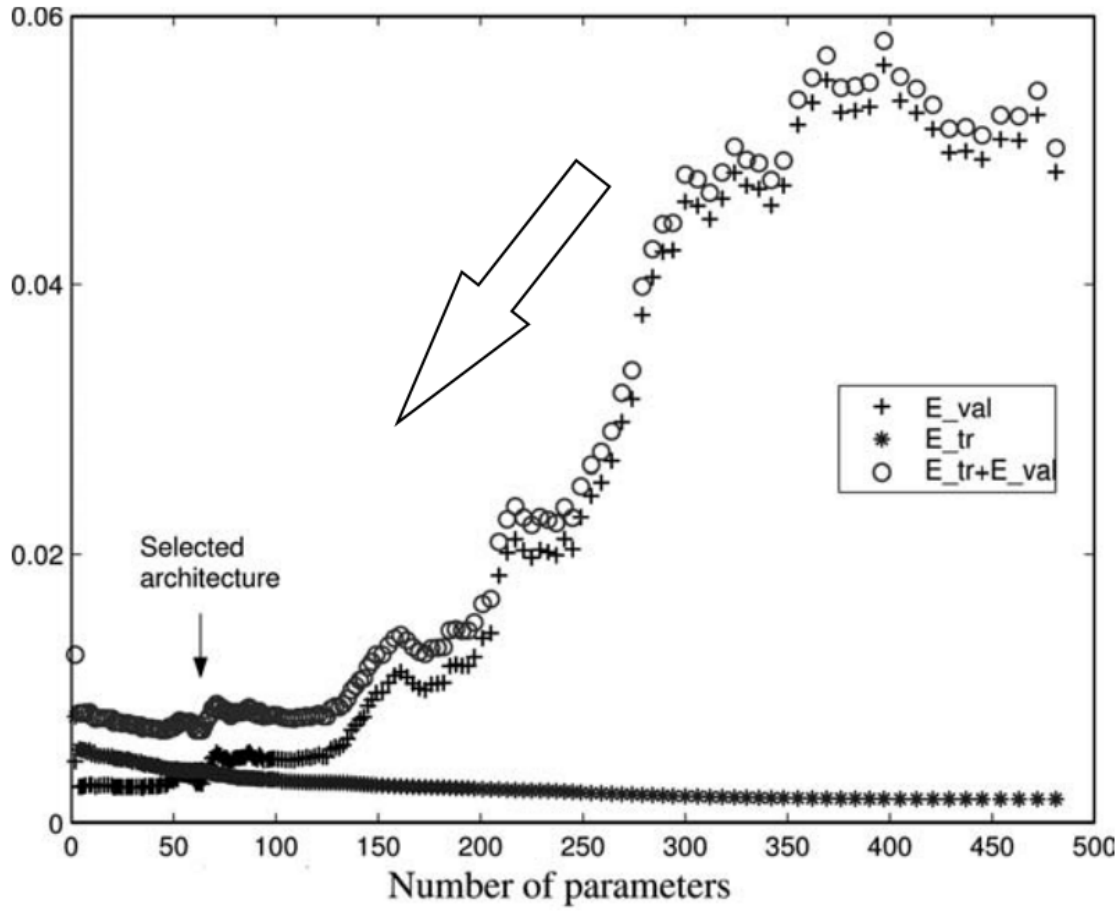


Figure 2.24. RMSE versus number of ANN parameters (Corani and Guariso, 2005)

Napolitano (2011), in her PhD thesis, examines the use of ANN model ensembles for the prediction of fluvial flooding in the Tiber river, Italy and Potomac river, USA. Various methods of combining the ensembles are trialled including use of partial information (PI) to select input features and use of a pre-processing technique: Empirical Mode Decomposition (EMD) (Huang et al., 1998) where individual time-series signals are modelled separately then recombined in an ensemble solution.

Tiwari and Chatterjee (2010b) use bagging – bootstrap resampling of the dataset with replacement – to build an ensemble of ANN models to predict hourly water levels at two locations in the Mahanadi river catchment in eastern India. The ensemble is also able to estimate uncertainties for the predictions. An attempt is made to build forecasts in hourly steps from 1 to 10-hours advance. The inputs consist of water level observations from 5 gauging stations on the river at a range of hourly lags. Selection of relevant inputs is effected by

means of three statistical techniques: cross correlation function (CCF), autocorrelation function (ACF), and partial autocorrelation function (PACF) between the input and the target variables, which are part of the data preparation pre-processing stage. Predictions are efficient above the 99% level for advances of up to 10-hours in this fourth-largest catchment in India. 95% confidence intervals on the predictions are also provided by the ensemble, which lends credibility to the results.

2.9.3 Bathing water quality

Chapter 5 describes case studies employing novel techniques for feature selection and visualisation of ANN weights, applying them to creating predictive models for bathing water quality. The case studies operate within the EU frameworks of the rBWD (European Commission, 2006a) and its predecessor (European Commission, 1976), which require monitoring of bacteriological counts from water at designated bathing beaches. These are then to be compared with threshold counts for two bacteria species and a classification of "fail" given based on exceedances of either count threshold; else "pass". In the event of exceedances, the public at the beach is to be advised that it may not be safe to swim. Therefore classifier models are required to make daily predictions for water quality safety at each designated beach, based on timely availability of certain environmental data, which are suitable to use as input features for the classifiers. The features used in our case studies are described in detail in Chapter 5.

This section reviews the work of other researchers to construct models for similar purposes. Particular focus is given to those using ANN and other machine learning techniques, but physically-based models are also briefly reviewed especially with regard to the types of input data they employ.

2.9.3.1 ANN models

ANN-based models for bathing water quality

Lin et al. (2008) employ coupled 2D and 1D hydrodynamic simulations of the flows and bacteria concentrations in the Ribble estuary, northern England. Results from these are used to calibrate an ANN model to predict bathing water

quality, using *faecal coliforms* as the measured organism. Figure 2.25 shows the case study area with sampling points marked.

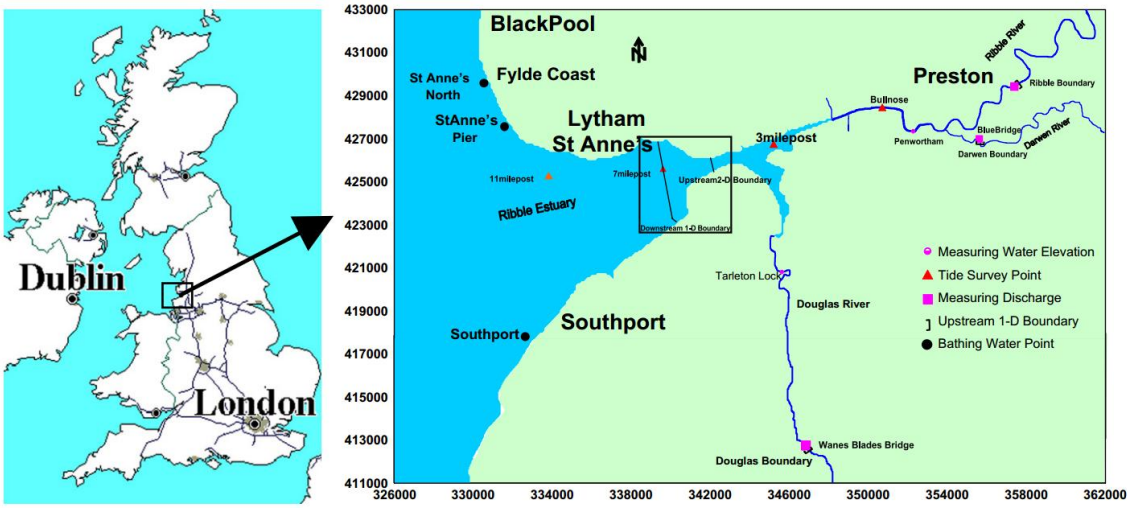


Figure 2.25. Ribble estuary case study area, showing sample points (Lin et al., 2008)

The use of the hydrodynamic simulations allows training and test data to be generated in far larger quantities than would have been possible using conventional water sampling, as well as modelling differing tidal conditions in dry and wet weather. The ANN inputs include flow and FC concentration from the Ribble, Darwen and Douglas rivers together with the corresponding water elevations and salinity levels. The ANN model output targets are the FC concentrations at the 7-mile and 11-mile post locations. The remaining inputs are salinity and water elevations at the target locations too. Feedforward 1HL ANNs are used, with the number of neurons on the hidden layer being determined by trial and error. Time lagged inputs of up to 18-hours are used meaning the number of inputs trialled is 8 (no lags), 27 or 53. The results show that generally performance is better for the greater number of inputs. However, if the flow regimes are taken into account, the number of inputs can be reduced without undue degradation in performance. The use of FC levels as inputs to the ANN models, however, would be problematic for a live real-time system, since these values would not be able to be supplied in a timely manner, due to the incubation time in the laboratory required.

Zhang et al. (2012) conduct a systematic, comparative study between three types of predictive model designed to meet the requirements of the US EPA BEACH Act. This requires beach managers to issue swimming advisories

when water quality standards are exceeded – as measured by *Enterococci* levels. All 3 models use the same datasets for the study. The data were collected and analyzed weekly during the bathing season (May to October) at six locations along Holly Beach, Louisiana, USA in the six year period from May 2005 to October 2010. Figure 2.26 shows the locations of the six sample sites.

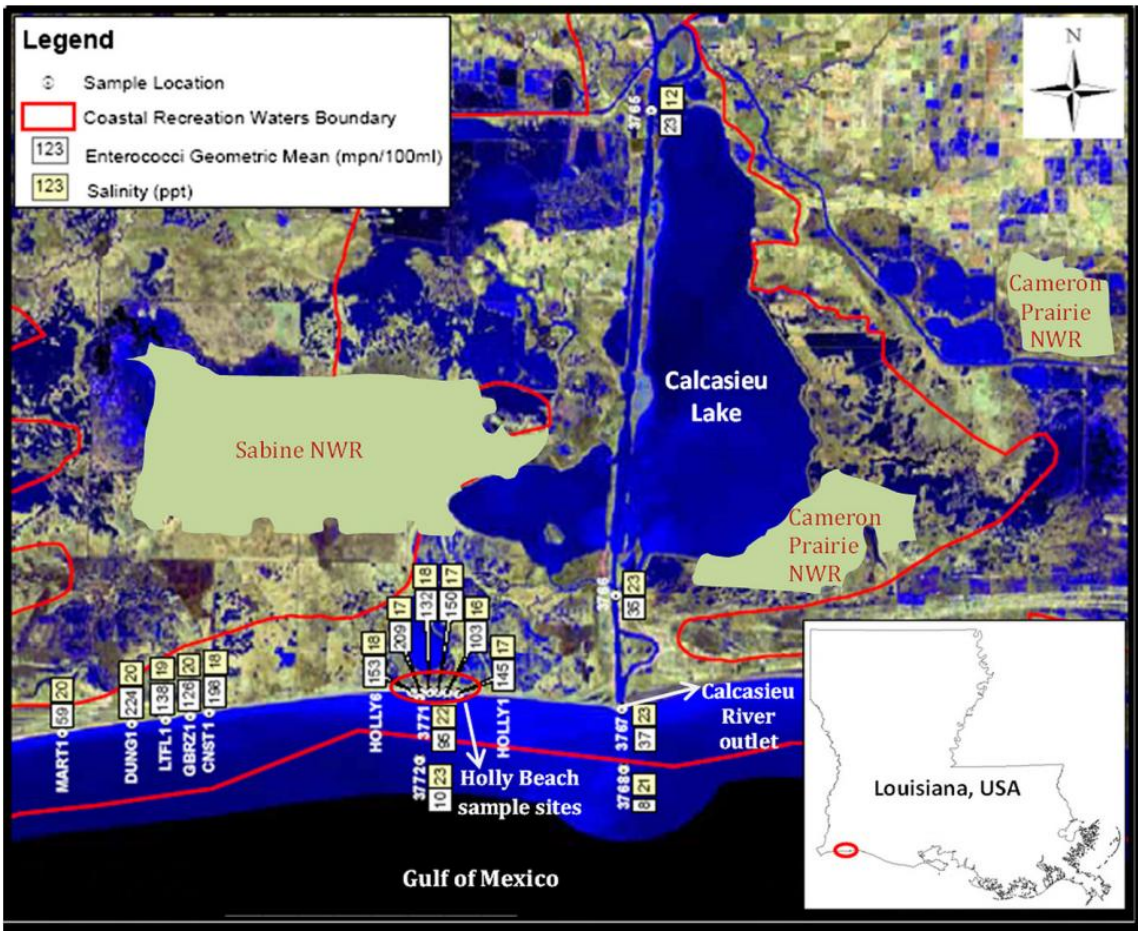


Figure 2.26. Location of Holly Beach, Louisiana sample sites (Zhang et al., 2012)

The ANN model includes 15 environmental variables including salinity, water temperature, wind speed and direction, tide level and type, weather type and various combinations of antecedent rainfalls. The other two models are generated using the Virtual Beach (VB) development platform and comprise a linear model and a model that includes non-linear transformations of some input variables.

A particular problem with the dataset is a continuous upward annual trend in the $log_e(Enterococci)$ levels of about 0.3 (log), the source of which is still unknown. However, this could have been solved by including timestamp as an input to the models.

The ANN model is a feedforward network with one hidden-layer of 5 neurons. There are 15 inputs and a single output predicting $log_e(ENT)$. Backpropagation is used as the training algorithm. Figure 2.27 shows the comparative results for the ANN (red), Linear VB model (blue) and non-linear VB model (mauve) against the observed $log_e(ENT)$ levels (green).

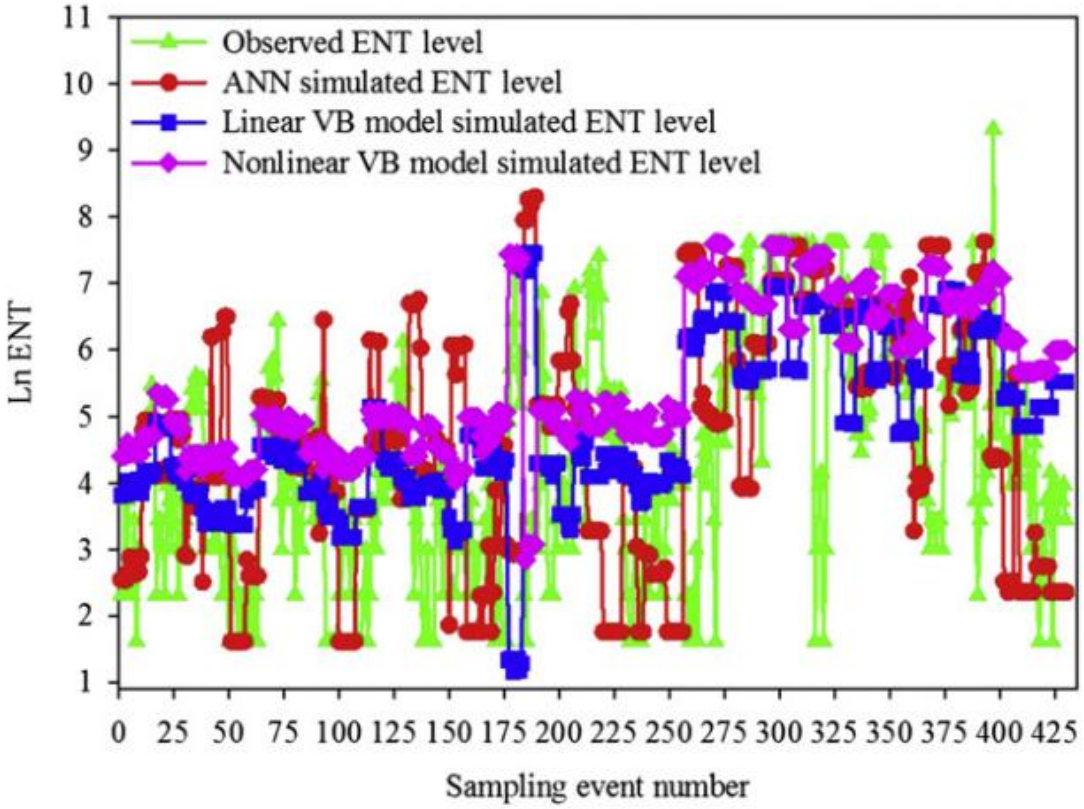


Figure 2.27. Comparison of observed and 3 model predictions @ Holly Beach (Zhang et al., 2012)

It can be seen that the ANN predictions are more responsive to the excursions in observed data than the other models – and this is confirmed by the numerical results for linear correlation coefficient and MSE.

Overall, this case study provides a useful comparison of ANN with conventional regression models and demonstrates generally better performance for the ANN. The study does not detail a regime for converting these results into advisories, nor, consequently, the misclassification error levels that would result from this. No mention is made of the optimisation of the architecture of the ANN. However, out of the papers reviewed, this study most closely matches with the work described in chapter 5 of this thesis.

Other ANN-based water quality models

Although the following studies are not specifically for bathing water quality, they are included here as the approaches taken to prediction of various water quality parameters are also potentially applicable to bathing water.

In a study to predict salinity in the Murray river, South Australia, up to 14-days ahead, researchers focus on selection of appropriate input features to build the best ANN-based model (Bowden et al., 2005). The method used is the measure of Mutual Information (MI) as a pre-processing step prior to selecting the relevant inputs to use for the model.

May et al. (2008) conduct a similar study using ANN-models to predict water quality in water distribution systems and use partial-mutual information (PMI) to select relevant input features for the ANNs.

He et al. (2011) use PMI as a method of selecting inputs for an ANN-based model to predict urban stormwater runoff quality, including parameters: turbidity, specific conductance, water temperature, pH, and dissolved oxygen (DO).

Aguilera et al. (2001) use Kohonen Neural Networks (KNN) – also referred to as Self Organising Maps (SOM) – to classify trophic levels of bathing waters into 4-classes potentially eutrophic; high mesotrophic; low mesotrophic; oligotrophic. The inputs are based on sampled levels of nutrients (ammonia, nitrite, nitrate and phosphate) at 22 locations along a 70km section of coastline. The case study area is a region of the south coast of Spain near to Almeria. The method is found to successfully classify previously unknown samples, following training.

Chen et al. (2004) compare three types of machine learning classifier models (ANN with 2-hidden layers, support-vector machine (SVM) and maximum likelihood (MLH)) in combination with reflectance data from Landsat TM satellite as model inputs. A 5-class scheme is implemented for water quality – with class 1 being turbidity-dominated and class 5 being chlorophyll-dominated. 88 water samples are taken on the same day (22 Dec 1998) over a wide area of the Pearl River estuary and Hong Kong and reflectance data in

spectral bands 1-4 are acquired from the satellite over the same area. 23 samples are retained for model testing and the remainder are used for training. The five optically active water quality parameters are sampled: turbidity (TURB), suspended sediments (SS), total volatile solid (TVS), chlorophyll-a (Chl-a) and phaeo-pigment (PHAE) and used to calibrate and test the models.

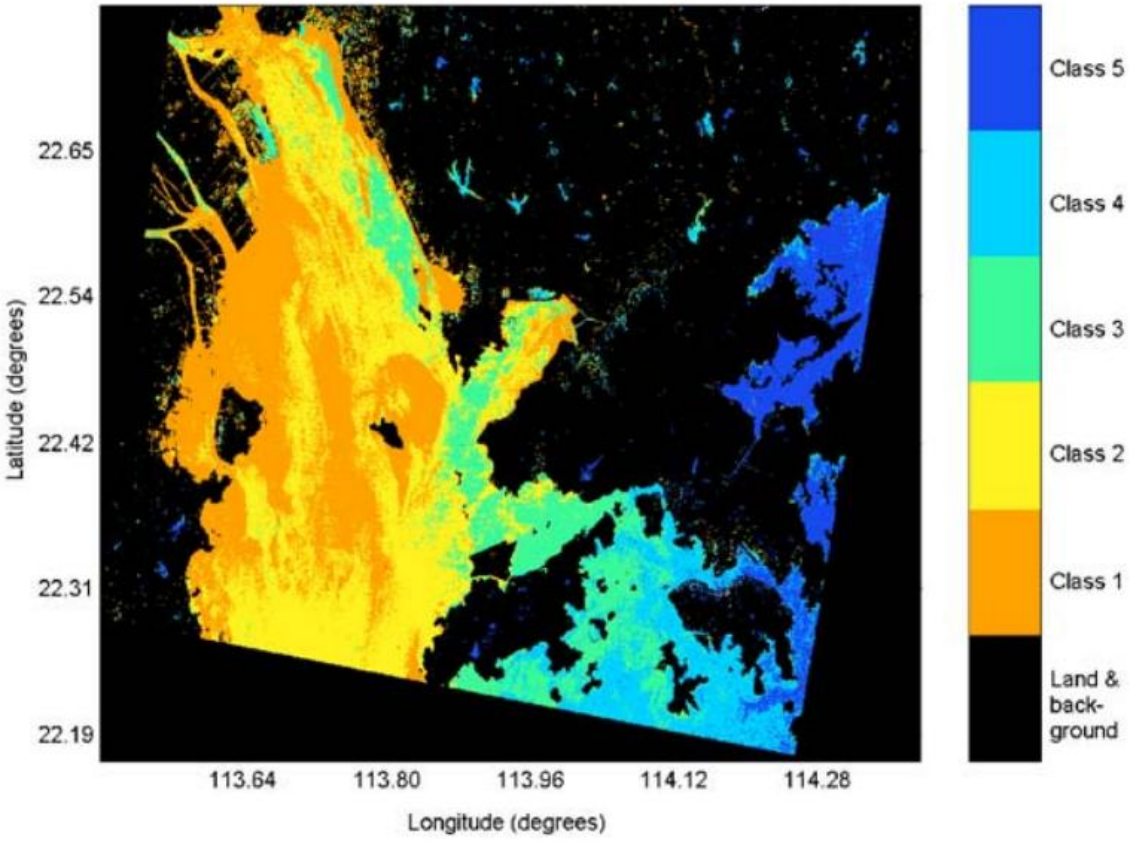


Figure 2.28. ANN classifications of Pearl river estuarine water quality (Chen et al., 2004)

Figure 2.28 illustrates the classification results for the ANN model, based on the spectral reflectance data processed from the Landsat TM images and calibrated against 65 samples of the 5 water quality parameters. Hong Kong Island can be seen at the bottom right of the image. The reported classification accuracies are: MLH: 78.3%; ANN: 82.6% and SVM: 91.3% based on the 23 test samples.

2.9.3.2 Other machine learning and/or data-driven models

Deng et al. (2012) present a decision support system (DSS) for managing and using recreational beaches. The DSS consists of:

1. A telemetered sensor-assisted water quality monitoring system
2. A multiple linear regression (MLR) model, developed with the VirtualBeach (VB) program, for predicting Enterococci levels, and
3. A web-enabled Geographic Information System (GIS) platform for displaying beach water quality.

The case study (and system) is designed and implemented for Holly Beach, Louisiana, USA and is known as “HollyBeachWatch”. Calibration is based on weekly *Enterococci* samples over a number of bathing seasons. The inputs for the regression model are seven environmental parameters including salinity, wind speed class, tide type, wind direction, 3 days antecedent rainfall, 48 hours antecedent rainfall and tidal water level. Variables can optionally be transformed using a number of non-linear functions if required – although this is not necessary in the finally selected model, given that the MLR system predicts *Ln(Enterococci count)*. The predicted values are post-processed to form “Advisory” or “No Advisory” classifications. “Advisory” means it is not safe to bathe. The VirtualBeach software provides a GA-based algorithm for selecting input features by building MLR models and comparing their performance. The selected MLR correctly predicts 88% of “Advisory” and 80% of “No Advisory” samples. Nowcasting and forecasting functions are also provided, although it is not clear what input data these are using. At the time of writing, the HollyBeachWatch system is not live.

Maimone et al. (2007) describe the development of a web-based forecasting system for bathing water quality for a non-tidal section of the Schuylkill river, Philadelphia. A three-class system is implemented: [green | yellow | red] using a manually generated and calibrated decision tree-like algorithm. Classifications are based on federal regulations for *E. coli*, although *faecal coliforms* are also measured in the calibration dataset. The input parameters are rainfall, river flow and turbidity, all of which are continuously monitored by telemetered gauges. The timestep is 1-hour. Although the algorithm could almost certainly have benefitted from a technique such as Genetic Programming (GP) or the use of decision trees (DT), which would have largely automated and optimised the calibration, the system benefits from a simple algorithm that has continued to work effectively over the last seven

years. At the time of writing, the system is currently live and a front-screen image is shown in Figure 2.29:

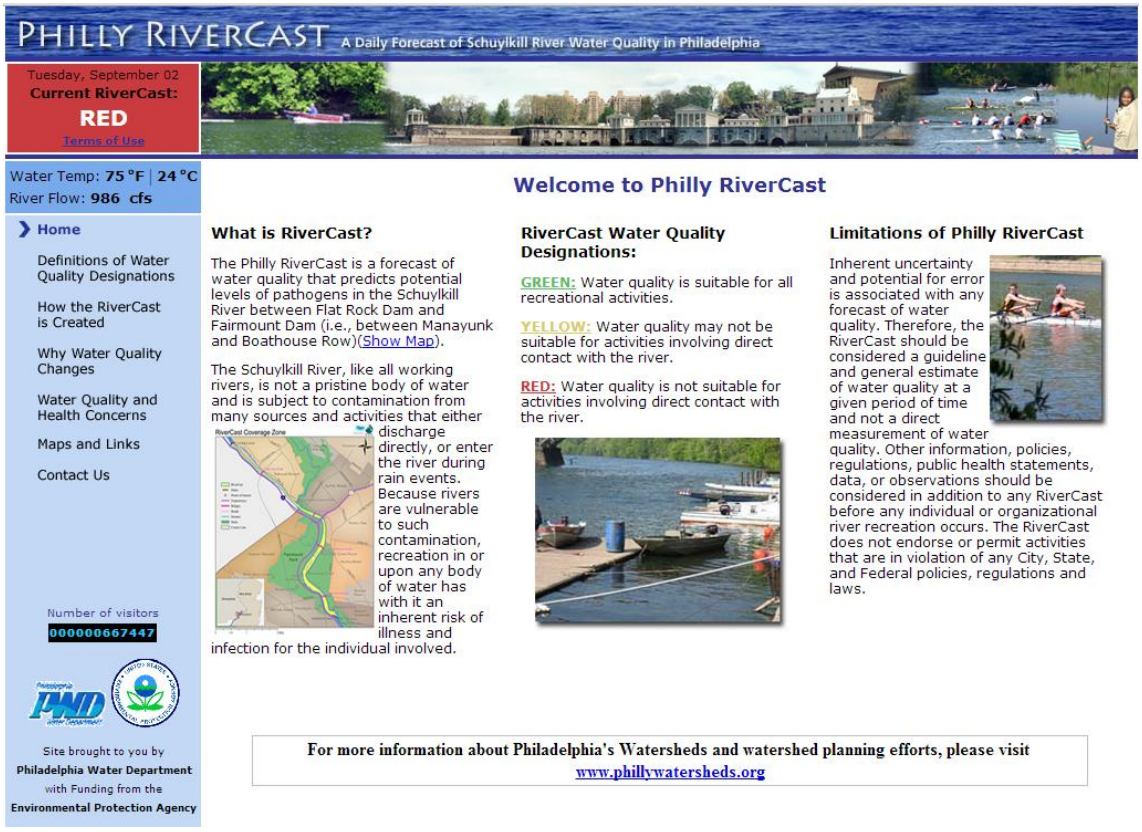


Figure 2.29. Philly RiverCast frontpage showing live water quality class = red (RiverCast, 2007)

2.9.3.3 Physical models

The COWAMA (Coastal Water Management) system (Suñer et al., 2007) is a system developed in Spain and trialled in Barcelona and Alicante in 2007 and in Barcelona, Biarritz and Sitges in 2008. It is based on hydrodynamic simulations and has two modes of operation: online and offline. The offline mode makes predictions of the overall beach classification based on historic data. The online mode uses both static and dynamic data. Static data inputs include details of the catchment and the urban drainage network(s) in the area. Dynamic data includes: rain gauges, water level gauges (sewer and/or river models validation in real time), current meters (hydrodynamic coastal model validation in real time), tidal gauges and pyranometer. The model includes components for each of the four subsystems of the water cycle: catchment, river, collection system, WWTP and receiving waters. The catchment model uses a semi-empirical method to distribute surface water to rivers and inlets to the sewer network. The river and sewer components implement the Saint

Venant equations in 1D. These predict water levels at ungauged locations. The WWTP model predicts CSO spills to receiving waters. The receiving waters component consists of a wave model and a 3D coastal flow model, which also computes transport and decay of pollutant concentrations; specifically *faecal coliforms* and *Escherichia coli* required for the rBWD (European Commission, 2006a). This project is an ambitious approach to producing reliable forecasts of bathing water quality at Mediterranean bathing beaches in accordance with the rBWD and is very much a physically-based modelling approach to this prediction problem.

A similar model is developed for prediction of bathing water quality (based on *E. coli*) at several bathing beaches in the neighbourhood of Hong Kong (Chan et al., 2013). Again a 3D deterministic hydrodynamic model of coastal flows is used. Of concern within the model is the Harbour Area Treatment Scheme (HATS) outfall discharging 1.4 million m^3/d of partially-treated sewage; the transport of this effluent is studied and modelled. The model achieves 81-91% overall accuracy in forecasting compliance / exceedance of the bathing water quality standard.

Twigt et al. (2011) describe a 3D model using the Deltares-FEWS system to model and predict algal blooms. The case study involves coastline in the Netherlands and the English Channel.

2.10 Summary of literature review

This chapter has reviewed ANN research and set it into context within machine learning and optimisation. Use of evolutionary algorithms to optimise ANNs ('Neuroevolution') has also been covered. Feature selection approaches have been discussed along with ensemble modelling techniques. Approaches to visualisation of ANNs – especially with regard to their weights, biases and/or neural pathways – have been reviewed. Finally, applications of ANNs in the areas of urban flooding and bathing water quality prediction have been reviewed and discussed.

Chapter 3: Case Study: Urban Flooding

This chapter describes case study work carried out with the objective of researching and improving machine-learning based predictive models for flooding from urban drainage networks in the UK. This work provides the motivation and basis for the development of the later models described in chapters 4 and 5, so is documented here to describe the research process leading up to the ANN ensemble and input feature-selection techniques described in the following two chapters.

3.1 Background

3.1.1 History of ANNs and DDMs

Considerable research on Artificial Neural Networks (ANNs) (McCulloch, 1943), (Rosenblatt, 1958), (Rumelhart and McClelland, 1986a), (Lapedes, 1987), (Hava T. Siegelmann, 1991), (Sontag, 1991), (Bishop, 1995) has been conducted. ANNs are examples of Data Driven Models (DDMs). Solomatine (2008) defines DDMs as follows:

"Data-driven modelling is based on the analysis of all the data characterising the system under study. A model can then be defined on the basis of connections between the system state variables (input, internal and output variables) with only a limited number of assumptions about the 'physical' behaviour of the system."

Research to date is extensively reviewed in chapter 2; the literature review. This covers ANNs and DDMs as well as predictive modelling for urban flooding. The reader is referred there for coverage of background and previous work.

3.1.2 Challenges of Urban Flooding

As referred to in an earlier paper co-authored by the author of this thesis (Savić et al., 2013); today, half of the world's population lives in cities and, by 2030, this will grow to nearly 60% (Heilig, 2012). The trends in urban population growth together with other pressures, such as climate change, create enormous

challenges to provision of resilient and safe urban drainage services despite, in many cases, ageing infrastructure. Urban drainage management involves consideration of sustainable use of water resources, pollution control, stormwater and wastewater network management and flood control and prevention. The high costs of expanding, renewing and strengthening the physical infrastructure to relieve these pressures mean there is a critical and urgent need to investigate and implement 'intelligent' management techniques toward improved use of the existing urban water infrastructure. This may help delay many large infrastructure investments otherwise required to mitigate urban flood-risks. In the UK, much of the older infrastructure is in the form of combined sewers, which inevitably means that floodwater becomes contaminated with sewage, with the associated health risks. These factors combine to make urban flooding an urgent urban management and planning issue.

"Intelligent grid" and/or "smart grid" are terms that have their origin in the electricity industry (Amin and Wollenberg, 2005). They refer to an electrical grid that uses information and communications technology (ICT) to automate processes that improve the efficiency, reliability, economics and sustainability of the production and distribution of electricity. This concept of smart-grid technology is being adopted in many countries around the world to ensure that electricity networks are flexible, accessible, reliable and economical (European Commission, 2006b). The intelligent grid concept will also benefit from the rapid increase in the amount of data (i.e., "big data") becoming available through proliferation of sensors, mobile communications, social media, etc. However, without intelligent computational methods, grid managers and decision makers will find it increasingly difficult to make sense of the large amount of data being made available in near real-time. In a similar vein to the smart electricity grid, "intelligent water networks" or "intelligent water infrastructure", which take advantage of the latest ICT to gather and act on information in an automated fashion, could allow the minimisation of waste and delivery of more sustainable water services. Additionally, they could help to mitigate the consequences of urban flooding through the provision of operationally useful predictive live real-time models.

This case study describes and analyses the use of ANNs as an example of machine learning-based intelligent systems developed to utilise increasingly available real-time sensor information in the urban water environment. It deals with urban drainage systems and utilises rainfall data to predict flooding for multiple urban locations in near real-time. Currently, observation data from urban drainage networks is still fairly sparse, but there is no need to wait for online monitoring "big-data" to become universally available; rapid real-time predictive models can be created and studied as data-driven surrogates of much slower and computationally demanding hydraulic or hydrodynamic models. These latter typically take rainfall hyetographs as input and produce flood level / volume / flow hydrographs for sewerage nodes as output, based on a parameterised physical model of a sewerage network and a set of physically-based equations describing the water flow into, through and out of the network (Zoppou, 2001).

Early Warning Systems (EWS), in order to be operationally useful, need to provide at least a 2-hour lead-time (Einfalt et al., 2004; Kellagher, 2012a). However, for large networks and/or when repetitive simulation runs are needed (i.e., for flood risk assessment), these can be slow and computationally expensive. A faster surrogate method based on Artificial Neural Networks (ANN) is presented here. This permits modelling of very large drainage networks in real-time, without unacceptable degradation of accuracy. It is worth noting that because these are not physical models, there is no need to model every sewerage node; it is sufficient to model only those nodes identified from the output of the physical model as having a probability of flooding above some threshold value: "key" nodes. Furthermore, in the case of the trained feedforward ANN's used in this study, there are no iterative loops; predictive outputs are obtained directly from a non-linear combination of time-lagged inputs. These two factors combine to produce considerable computational cost saving for the models, once trained, and hence speed improvement when compared to physically-based hydrodynamic models.

3.1.3 Artificial Neural Networks for Urban Flood Modelling

As part of University of Exeter’s contribution to research under the Flood Risk Management Research Consortium Phase 2 project (FRMRC2, 2011;

Schellart et al., 2011) and the UK Water Industry Research (UKWIR, 2012) follow-on case studies, the "RAdar Pluvial flooding Identification for Drainage System" (RAPIDS) is developed using a single, multi-output ANN to predict flooding at multiple nodes in sewer systems (Duncan et al., 2011, 2013a, 2013b). This approach exploits the similarities between hydrographs at different sewer nodes, which make the modelling of an entire sewer network by a single multi-output ANN feasible; a contribution made by the author and a technique described in detail in this chapter.

This case study assesses the opportunities provided by data-driven ANN-based models for rapid and concurrent predictions of urban flooding from manholes and Combined Sewer Overflow (CSO) spills at multiple locations in a sewerage network. This could provide water utilities and local authorities with the ability to improve their level of service and compliance with regulation as well as reduce risks to their customers and the general public, through taking effective action to mitigate impacts of flooding.

A sensitivity analysis is also described here that looks at the limits of predictive ability of time-lagged ANN models, when based on actual rainfall as input. This shows that prediction advance is broadly limited to the Time of Concentration (ToC) for each node. This can be approximated by the delay of the peak of cross-correlation between the rainfall input hyetograph and the hydrograph for each given node in the sewer network, when measured for the longest duration rainfall events. ToC describes the maximum transit time of water from the furthest (upstream) point of the urban catchment to the given node (Butler and Davies, 2004). With the exception of the most downstream nodes in the very largest urban drainage networks, this would normally be very short, i.e., of the order of tens of minutes, rather than hours, thus requiring prediction of rainfall to achieve the required operational lead-times.

Rainfall nowcasting (forecasting <6 hours ahead) is commonly obtained from radar rainfall images (Schellart et al., 2009; Wang et al., 2009). Although work has been carried out with the UK Met Office Nimrod 1km composite radar images (with 5-minute temporal resolution) and Environment Agency telemetered raingauge network (with 15-minute temporal resolution) (UKWIR, 2012), in this study results are presented both based on synthetic design rainfall

using a range of return periods and durations as well as real (actual as opposed to predictions of) rainfall events for three urban catchments of different sizes, located in southern England.

Due to the lack of measured data from urban flooding events for the case-study urban drainage networks, the InfoWorks CS model (Innovyze, 2012) is used to provide time-series (hydrographs) describing system performance at manholes, CSOs and outfalls. ANN models are then developed to predict performance at these key points of interest for any rainfall loading condition and these predictions are compared to InfoWorks CS results, which are treated as 'ground truth' for the purposes of the study. During training of the ANN model, they represent the target signals and during system test they represent reference signals against which to evaluate the predictive error.

In the context of urban drainage modelling and flood prediction, a relationship exists between the possible input variables (including antecedent, current and predicted rainfall, soil moisture conditions) and the resultant hydrographs of what can be thought of as the outputs of the system – depths, flow rates and volumes at manholes, outfalls, CSOs (Combined Sewer Overflows) etc. These are described in more detail in section 3.6.1.1.

The DDM training procedure can be seen as calibration or optimisation of the model to characterise that relationship and to minimise the difference between observed hydrographs from the sewer network and those produced by the model, for the same set of input conditions. *Note that this can be done without expression of that relationship in terms of mathematical equations describing underlying physical laws.* Instead, the transfer function between the input data set and the desired target output data set is "learnt" through adjustment of the collection of multiplicative weights and offset biases for each layer of neurons in the network in a strategy to minimise the difference between the ANN output during training and the target training data set (e.g. flood hydrographs).

The premise for this approach to be valid is the system state (the network) must remain the same. Thus systems which are altered (different pipe sizes / areas allocated / Real Time Control (rules controlling hydraulic structures) will

result in ANN models having limited or no success in predicting the performance at any point depending on the degree and location of the change(s) in the system state. This limitation is generally not problematic, since it is reasonable to consider recalibration of models in the event that the modelled sewerage network is modified. It may be possible, using additional input information, or using multiple models for ANNs to be used for multiple system states, but this is not in the scope of this study.

3.1.3.1 High Dimensionality & Strategies for Dimension Reduction

A problem reported in machine learning and pattern recognition literature is the so called "*Curse of Dimensionality*" (Bernecker et al., 2011), (Chávez et al., 2001), (Houle et al., 2010)

In the case of modelling with ANNs, there is a law of diminishing returns in relation to the number of inputs to the ANN. The number of dimensions in the search space for the optimum solution or model is a positive function of the number of inputs to the ANN, since there is a 1: N relationship between inputs and network input-layer adjustable weights, where N is the number of neurons on the hidden layer. As the number of inputs increases (as is the case, for example, with spatial rainfall and/or spatially variable New Antecedent Precipitation Index (NAPI) (Kellagher, 2012b) values), the number of rainfall event samples required in the training data set and the number of candidate solutions needed to explore the search space of network weights increases exponentially. This has been widely reported as the Curse of Dimensionality.

This is a problem that also affects the modelling of multi-nodal sewer networks for urban flooding, since there may be spatially and temporally varying rainfall and soil moisture conditions (NAPI)(Kellagher, 2012b), yielding possibilities for many hundreds of input features to the models.

Strategies for dimension-reduction include: feature extraction, selection by relevance (Factor Analysis)(Harman, 1960), Principal Component Analysis (PCA)(Jolliffe, 2005), Independent Component Analysis (ICA)(Roberts and Everson, 2001) and Multi-Unit objective Functions and Random Projections (Fodor, 2002). Further research is needed in relation to the application of these

techniques to urban flood modelling when using spatially and temporally variable input data, such as rainfall and NAPI. Chapters 4 and 5 detail a novel approach to dimension reduction, named “Neural Pathway Strength Feature Selection (NPSFS)”, developed by the author as a result of work on predictive models for urban flooding (this chapter) and bathing water quality (chapter 5).

3.2 Overview of ANN techniques used

An overview of ANNs in general is provided in the literature review in chapter 2 so is not repeated here; but a brief review follows:

Architectures include fully-connected and layered networks. This study uses feed-forward Multi-Layer Perceptrons (MLPs). These are layered and process data uni-directionally from input to output. Figure 3.1 illustrates 3-layered feed-forward ANN, which is fully-connected within each layer. Note: the input layer simply distributes inputs to all neurons in the hidden layer; there are only 2 layers of neurons.

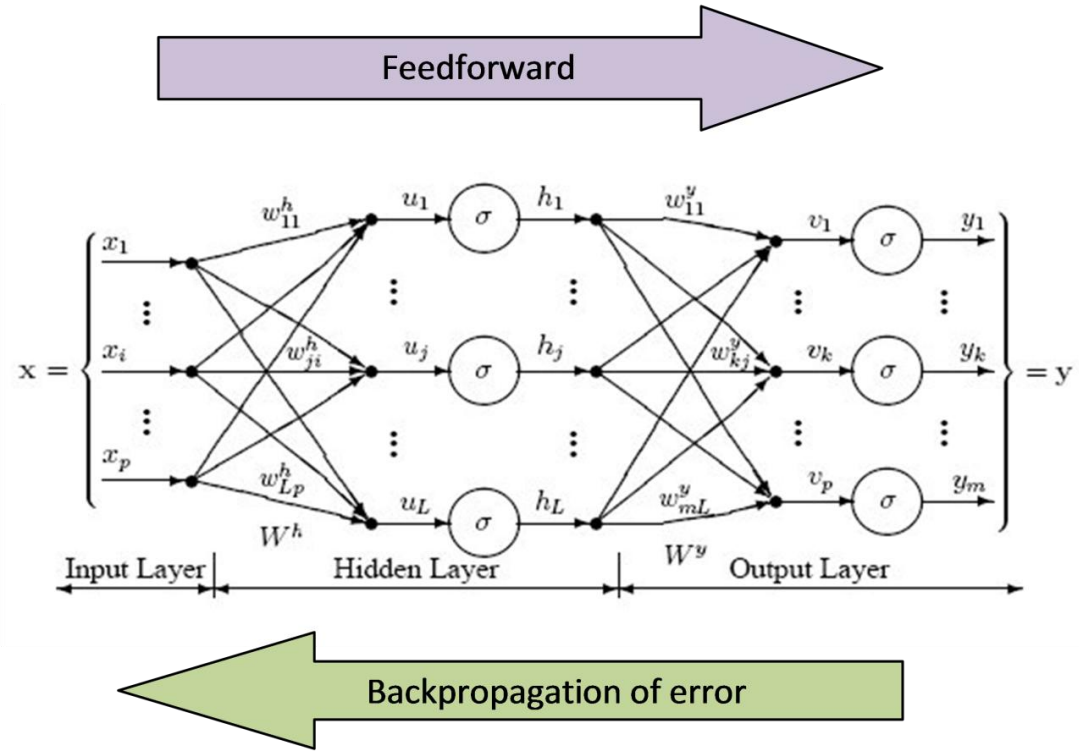


Figure 3.1. Typical Multi-Layer Perceptron Architecture (Public Domain image)

In the ANNs used, the hidden layer neurons utilise a tanh (i.e. non-linear, sigmoid) activation function. Because the production of hydrographs involves regression rather than classification or probability estimation, the output neurons use a linear activation function. The use of non-linear activation

functions permits the network to model non-linear relationships between the input and output data sets. Barron (1993) shows that all continuously differentiable finite functions can be approximated with an error *O(1/n)*, where *n* is the number of copies of a superposed sigmoid function. This is important for our application and is implemented by the layer of hidden neurons.

Supervised training is used, in which expected target data are known for a given set of input data (e.g., rainfall). Target data (e.g., manhole water depths produced by an InfoWorks CS model, in this case study) are compared to the output generated by ANN and errors back-propagated towards the input, adjusting weights so as to reduce the output error. Error optimisation strategies include Scaled-Conjugate-Gradients (SCG) and Quasi-Newton, both of which are gradient-based. Research to investigate alternative strategies, using alternatives to gradient-based methods, including Evolutionary Algorithms (EA)(Deb et al., 2002a; Zitzler et al., 2000) is described in chapter 5, but is not undertaken in this chapter. Other techniques that could be researched would include Particle Swarm Optimisation (PSO) (Kennedy and Eberhart, 1995) and Ant Colony Optimisation (ACO) (Dorigo and Blum, 2005).

3.3 Case study project stages

This case study is carried out in 3 stages; the first two of which were included in the UKWIR RTM project. The Design Rainfall Experiment/Stage tests the effectiveness of ANN models on “simple” design rainfall data. The Real Rainfall Experiment/Stage applies ANN models to more complex and realistic data. Finally, in the Sensitivity Analysis Stage, the limits of prediction achievable by a DDM, such as an ANN, when based on actual rainfall as input are researched.

In the first stage trials, 16 design rainfall events of various return periods and durations are used for each case study catchment. Fixed durations are used for all events, by varying the runoff periods to match the given duration of rainfall. Spatially uniform rainfall is used for all catchments.

For the Real Rainfall Experiment/Stage, selections of 50 real rainfall events are used for each case study catchment. Durations vary between

approximately 6 hours and 100 hours dependent upon event and catchment. Spatially variable rainfall is used for one (largest) catchment; whereas the others use spatially uniform rainfall.

The Sensitivity Analysis stage also uses the design rainfall events, as their single-peaked nature makes the analysis much easier to perform.

3.4 Objectives of case studies

The objectives of this study are:

- To research the use (as a rapid surrogate for InfoWorks CS) of a Data-Driven-Model (DDM), which treats the sewer network as a black box;

- To demonstrate and evaluate the capability of the ANN models to predict sewer flooding for 3 different urban catchments located in southern England, for both real and design rainfall events;

- To model relationships between inputs and outputs using feedforward ANNs with time-lagged inputs in moving time windows;

- To include modelling of higher-order interactions between inputs and between the various outputs;

- To evaluate the ANNs as regression models to predict continuous quantities;

- To include and evaluate the performance of a post-processor / wrapper function to classify flooding using two alternative schemas;

- To demonstrate limits of the ability to predict potential flooding (for durations up to the flow travel time (Time of Concentration) within the sewerage system, when using actual (rather than predictions of) rainfall.

3.5 Case study catchments

The following 3 figures are Google Earth® aerial views of the three catchments, which have been used for this case study, with the drainage systems overlaid: Figure 3.2 depicts the Crossness sewer network; from which 39 flooding (blue circles) or surcharged nodes (orange circles)(section 3.6.1.1) are selected for modelling. Similarly, Figure 3.3 shows the Dorchester urban catchment and Figure 3.4 displays the Portsmouth catchment.

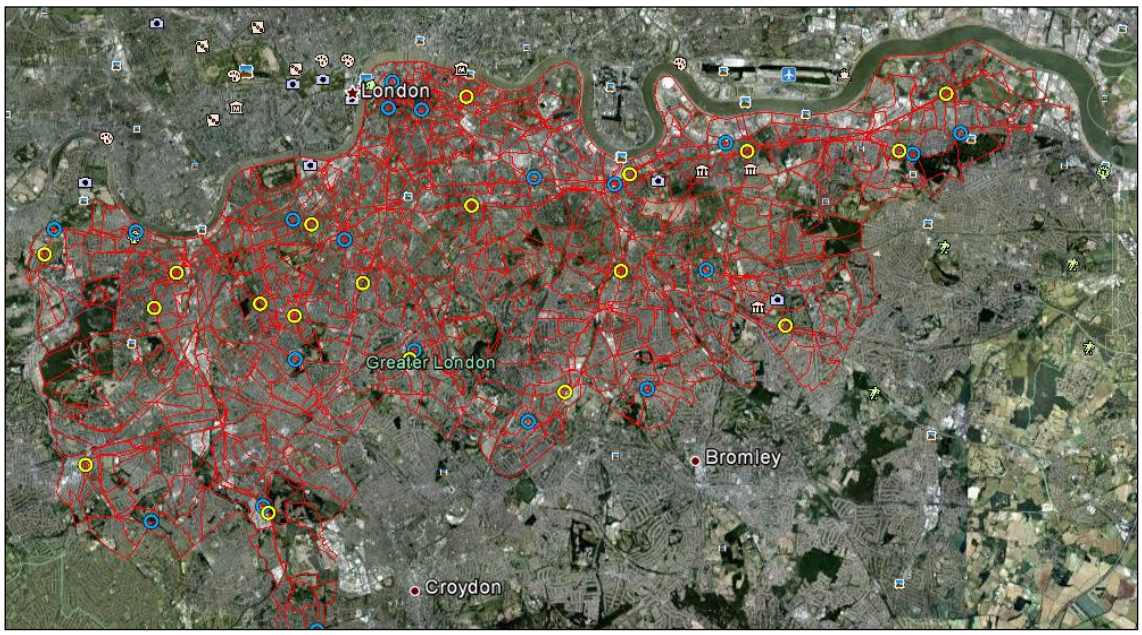


Figure 3.2. Google Earth® image of southern Greater London with modelled Crossness network

Key: Orange – surcharged manholes; Blue – flooding manholes

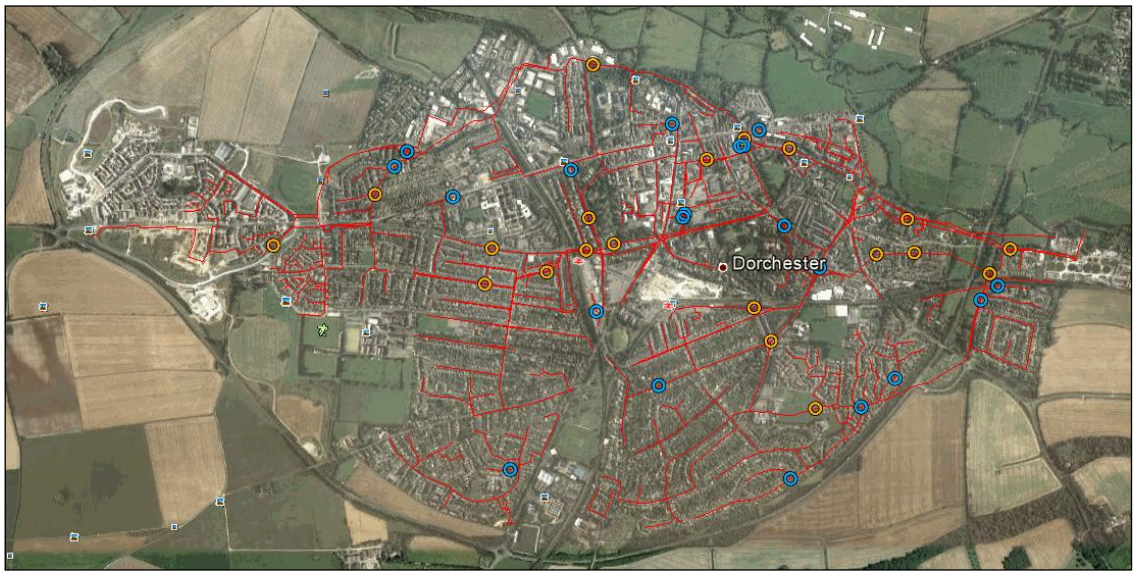


Figure 3.3. Google Earth® image of Dorchester, UK, with modelled sewer network

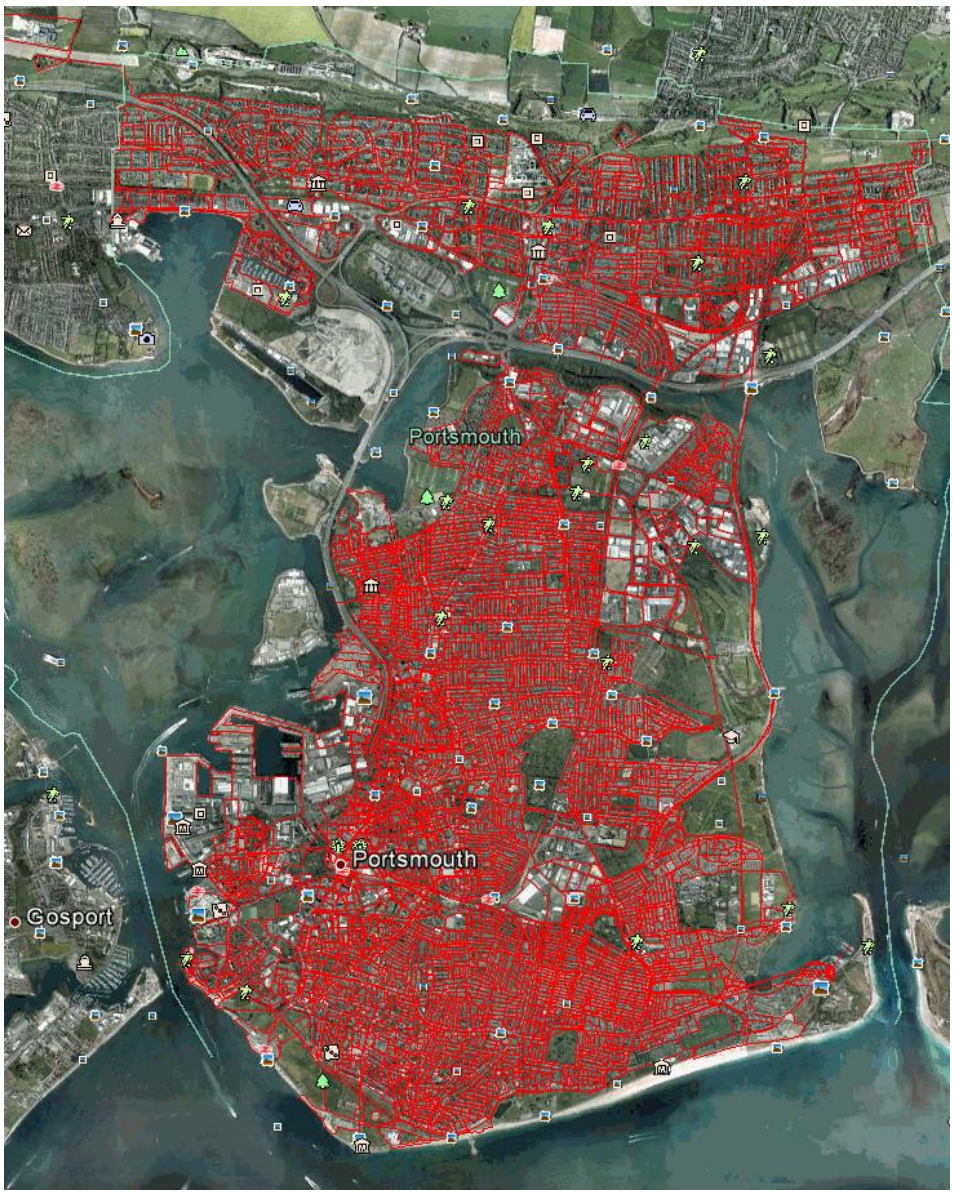


Figure 3.4. Google Earth® image of Portsmouth, UK, with modelled sewer network

Nodes in a drainage network are points at which pipes/conduits connect; where there is usually a manhole or more rarely a CSO (section 3.6.1.1). Specifically, for the indicated nodes, flooding or surcharge is found to occur for at least one of the extreme (design) rainfall events modelled in the first phase of the project. These nodes are then also used for the second phase with real rainfall.

The three catchments are selected to represent a range of scales, from the smallest to the largest: Dorchester (6.9km^2); Portsmouth (29.6km^2) and Crossness (230km^2).

The following three tables present the characteristics of each catchment:

Table 3.1. Crossness catchment network overview (UKWIR, 2012)

Attribute	Value	Units
Nodes Total	2709	
Nodes Manholes	2676	
Nodes Outfall	33	
Combined Sewer Overflows (CSOs)	29	
Pipes	2600	
Orifice Fixed	30	
Pump Fixed	1	
Pump Screw	23	
Pump Roto Dynamic	5	
Sluice Fixed	45	
Sluice Variable	10	
Weir Fixed	168	
Flap Valves	37	
Head Discharge Curves	62	
Pipe Length	426133.3	m
Pipe Size	152 - 5715	mm
Subcatchments	436	
Subcatchments Total Area	22999.09	ha
Subcatchments Contributing Area	22999.09	ha
Subcatchments Population Count	1813381	
Subcatchments Runoff Surface	16217	ha
Subcatchments Runoff Surfaces - Impervious	60	%
Subcatchments Runoff Surfaces - Pervious	39	%
Subcatchments Land Use Profiles	7	
Subcatchments Wastewater Profiles	4	
Rainfall Profiles (raingauges)	23	

Figure 3.5. Crossness catchment showing Thiessen polygons for rainfall profiles (UKWIR, 2012)

Table 3.2. Dorchester catchment network overview (UKWIR, 2012)

Attribute	Value	Units
Nodes Total	1391	
Nodes Manholes	1365	
Nodes Outfalls	20	
Combined Sewer Overflows (CSOs)	10	
Pipes	1377	
Orifice Fixed	2	
Pump Fixed	8	
Weir Fixed	17	
Flap Valves	4	
Pipe Length	61152.54	m
Pipe Size	100-2000	mm
Subcatchments	773	
Subcatchments Total Area	692.99	ha
Subcatchments Population Count	19824	
Subcatchments Contributing Area	692.99	ha
Subcatchments Runoff Surfaces - Impervious	12	%
Subcatchments Runoff Surfaces - Pervious	87	%
Subcatchments Land Use Profiles	2	
Subcatchments Waste Water Profiles	3	
Rainfall Profiles (raingauges)	1	

Table 3.3. Portsmouth catchment network overview (UKWIR, 2012)

Attribute	Value	Units
Nodes Total	8546	
Outfalls	21	
Combined Sewer Overflows (CSOs)	16	
Pipes	8596	
Orifice Fixed	18	
Pump Fixed	64	
Pump Roto Dynamic	5	
Sluice Fixed	8	
Sluice Variable	11	
Weir Fixed	73	
Flap Valves	14	
SOIL type	3	
Subcatchments Total Area	29.6	km^2
Subcatchments Contributing Area	29.6	km^2
Subcatchments Population Count	188060	
Subcatchments Runoff Surface	23.4	km^2
Subcatchments Runoff Surfaces - Impervious	45	%
Subcatchments Runoff Surfaces - % Pervious	55	
No. of Subcatchments Land Use Profiles	2	
No. of Subcatchments Wastewater Profiles	1	
Rainfall Profiles (raingauges)	1	

3.6 Methodology

3.6.1 The selected structure of the ANN models

As stated in section 3.2, the ANN's used are 3-layer fully-connected, feed-forward Multi-Layer Perceptron (MLP) networks, in which the input layer units merely distribute all inputs to all neurons in the hidden (second) layer.

3.6.1.1 Types of sewer node and quantity predicted

The three types of sewer node modelled are:

Manholes; these are junction points between pipes in an urban drainage network. In combined sewers, they are also connected to gully pots, which take surface runoff water from the street gutters. There are three significant water levels (see Figure 3.6). "Depths" are measured with respect to the cover level (ground surface level), so negative values mean water surface levels below the cover. Positive values indicate flooding with the water surface above cover. Soffit level is the level of the top of the outlet pipe. When water level in the manhole is above this level, the manhole is said to be "surcharged", because the outlet pipe is running at 100% of capacity and is pressurised. Finally, for the purpose of this study a water level 1 metre below cover is taken as "cellar flood level", since backflows from pipes above this level can occur into cellars of properties nearby.

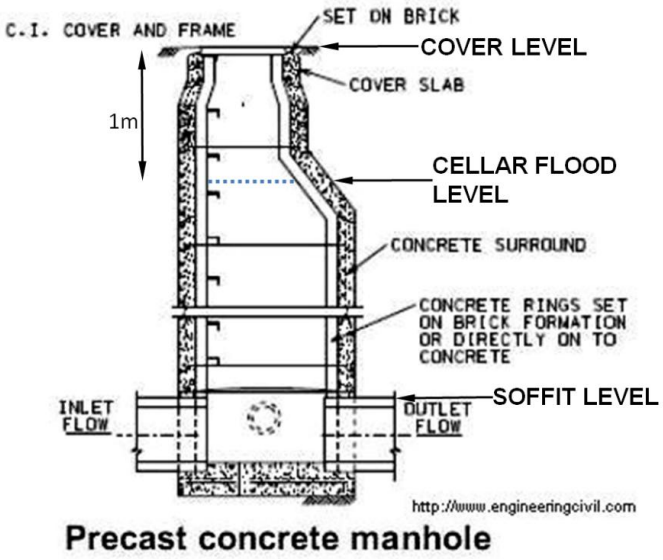


Figure 3.6. Cross-section of sewer manhole showing levels (Chu, 2007)

- Combined Sewer Overflows; these are safety devices fitted within combined sewer networks just upstream of the sewage treatment works (STW) to prevent the STW from being inundated by peak flows during rainfall events. In the event of the chamber (front of Figure 3.7) becoming full, raw untreated sewage mixed with flood-water[12] spills over the weir and is allowed to run off to a receiving water (usually a river). This is undesirable, but less problematic than inundation of the STW. "Depths" in the CSO are taken with respect to the top of the weir, with positive values indicating that a spill is occurring.

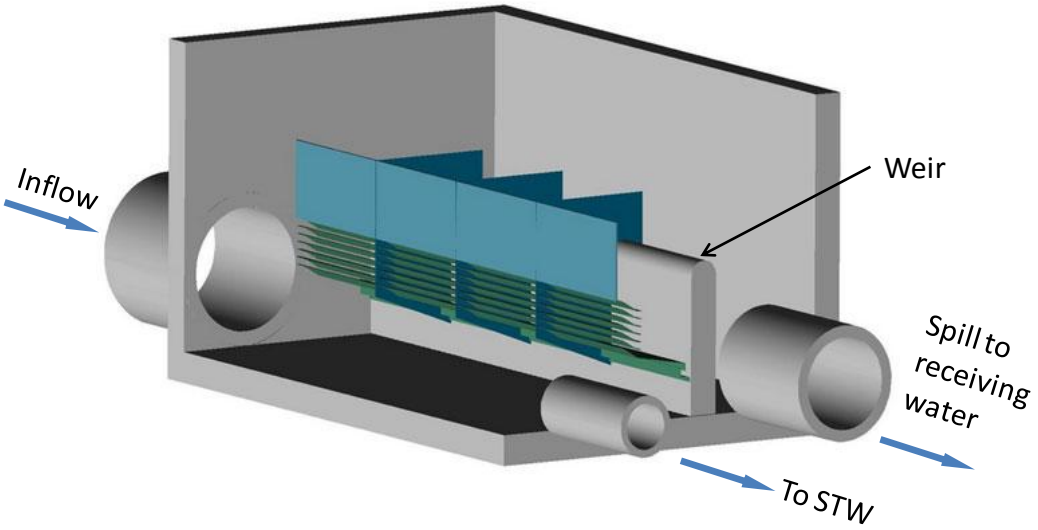


Figure 3.7. Combined sewer overflow (CSO) showing overflow weir (Biogest, 2014)

- Outfalls; these are the outflow pipes at the output of an entire sewer network and would normally discharge the treated effluent from the STW to a receiving water. "Depths" are taken as above the "invert" level of the bottom of the pipe at the outfall.

The three quantities predicted by the models are:

- Water level (also referred to as "depth") measured in metres; for manholes, this is with respect to the cover; for CSOs it is with respect to the weir and for outfalls it is with respect to the invert of the pipe.
- Flow rate measured in cubic metres per second (m^3s^{-1}); for manholes, this is the rate of flow from the cover (in the event of flooding); for CSOs it is flow

[12] *The illustration shows preliminary lamellar screening (blue) that traps the larger solids in the event of a spill and prevents them reaching the receiving water.*

rate over the weir (in the event of a spill) and for outfalls it is the flow rate from the outfall.

- Volume measured in cubic metres per timestep (m^3t^{-1}); for manholes, this is the volume of flow from the cover per timestep (in the event of flooding); for CSOs it is volume over the weir per timestep (in the event of a spill) and for outfalls it is the volume from the outfall per timestep.

Early work on the ANN models demonstrated that best results are obtained if all outputs being modelled by a single multi-output unit ANN are of the same type. Mixing types of quantity predicted yielded poor results. Also the very best results are obtained if the sewer node types are also kept the same within each ANN model. Therefore the results reported follow these guidelines unless otherwise stated.

3.6.1.2 Number of input units and timesteps

As is established practice when using ANNs for modelling using time-series data, lagged inputs are used in a moving time window of a certain number of timesteps wide (Dawson and Wilby, 2001; Luk et al., 2000; Maier and Dandy, 2000; Napolitano, 2011). This consists of all timesteps in a window $X_t = \{x_t, x_{t-1}, x_{t-2} \ldots x_{t-w}\}$, where X_t is the vector of inputs at timestep t; $w+1$ is the number of timesteps in the lagged moving time window.

The number of input units is given by the number of input signals *[rainfall intensities* | *cumulative rainfalls]* plus optionally *[NAPIs* | *pump states* | *tide levels]* multiplied by the number of timesteps in the moving input time window. This can quickly sum to a large number of input features. Many authors have used a variety of strategies and algorithms to reduce the number of input features and these are discussed in more detail in chapters 4 and 5 as well as in the literature review in chapter 2.

The results of this study have shown that using the elapsed time since start of a rainfall event as an input feature (linear ramp starting from zero at the beginning of each event) is unnecessary and can be omitted with negligible effect on ANN outputs.

For Dorchester and Portsmouth catchments, for both the design rainfall and real rainfall experimental stages, spatially uniform rainfall is represented by a single rainfall intensity hyetograph and single cumulative rainfall signal. Both catchments also optionally used a single spatially uniform New Antecedent Precipitation Index (NAPI) signal as a measure of soil moisture. Portsmouth is also used for the Sensitivity Analysis stage, again with spatially uniform rainfall.

For Crossness, uniform design rainfall is adopted for the first stage. For the Real Rainfall Experiment/Stage, spatially varying rainfall and NAPI signals are used. Rainfall is represented by data from the 23 raingauges covering the area of the Crossness catchment (Figure 3.5). These are described in the UKWIR RTM project report (UKWIR, 2012). Each raingauge provides 2 signals; rainfall intensity and a computed resultant cumulative rainfall for each event, giving a total of 46 rainfall signals. Spatially varying NAPI is also employed. NAPI traces from locations (nodes) in the catchment provided a further 40 input signals. In total (including a single elapsed time signal) 87 input signals are initially employed.

Based on early trials, a standardised moving input time window of 10 timesteps (i.e. 9 intervals) is selected in order to demonstrate that reasonable results could be obtained for different catchments using as standard ANN architecture as possible. The selection of 10 timesteps is based on the trade-off between obtaining reasonable results and the most parsimonious network. Results in this chapter and Shere (2012) suggest that the approach may benefit from using a slightly longer time window, dependent on delays inherent in the catchment.

Dorchester and Portsmouth models employ a 2-minute timestep, giving an 18-minute input time window. The Crossness model employs a 5-minute timestep, giving a 45-minute window.

In accordance with standard practice, all input signals are normalised to a range of [0, 1] prior to application to the ANN, in order to ensure that hidden layer input weight values would be close to the range [-1, +1] whilst making use of the non-linear portion of the sigmoid activation functions as necessary.

3.6.1.3 Number of Hidden Units

The number of hidden units is selected, based on early trials, again to provide the most parsimonious network whilst achieving generally acceptable results. Wherever possible, 10 hidden units are used as standard throughout both the design rainfall and real rainfall experiment/stages. The selection of too few hidden units can be associated with "underfitting", which means that the structure relating the input data to the target data is not able to be fully expressed by the network. The presence of too many hidden units can lead to "overfitting", a problem caused by the network learning from the noise in the training data set as well as the signal. This results in poor ability of the network to generalise to new rainfall events, causing sub-optimal ANN performance during evaluation of the test events. Since there are 3 or 4 signals multiplied by 10 input timesteps = 30 or 40 ANN inputs, there is a ratio between inputs and hidden units of either 3 or 4.

The one exception to the use of 10 hidden units is for Crossness in the Real Rainfall Experiment/Stage, where spatially variable NAPI and rainfall are provided as inputs to the ANN. Here, 100 hidden units are found to perform better. This figure is selected as a compromise, based on there being 87 signals times 10 input timesteps = 870 ANN inputs, leaving a ratio between inputs and hidden units of 8.7. As discussed in section 3.1.3.1, this amount of data leads to too high a level of dimensionality in the decision (weight) space of the ANN for an optimum solution to be located given the number of training event samples available.

This extremely complex model was expected to perform poorly, based on results reported in the literature, prior to starting work on it. However, it is common for urban drainage networks to approach the size of the Crossness catchment and to require modelling with spatially varying rainfall. This has motivated the further research described in chapters 4 and 5 to investigate possibilities for selection of the spatially variable inputs, based on their relevance for each sewerage node, so as to reduce the problem to a level of dimensionality (in terms of number of network weights) that would be technically feasible to calibrate successfully. Due to the size of the Crossness catchment, it

has not been possible to perform this calibration within the scope of this thesis and this is left as a proposal for future research.

3.6.1.4 Additional ANN configuration parameters

A number of other ANN configuration and setup parameters are selected as follows[13]:

Hidden layer activation function: *tanh* (which implements a symmetrical sigmoidal hyperbolic tangent transfer function between the limits of [-1... +1] for each hidden unit) is selected. This ensures the ability of the networks to model non-linear relationships between inputs and target hydrographs (Barron, 1993).

Output units: The number of output units is always equivalent to the number of sewerage nodes being modelled, given by the number of columns of target hydrograph data there are in the input data files.

Output layer activation function: a *linear* function is implemented: i.e. the identity function (x1) for the output of the summation process for each output neuron. This ensures that hydrographs can be modelled directly, based on the raw amplitudes and units of the target hydrographs. This is standard practice for the application of ANNs to regression problems.

Training function: *SCG* scaled conjugate gradients based optimisation algorithm is selected following evaluation of a number of alternatives early during the project development. It demonstrates robust performance for a variety of models and training data sets. The training data are presented and weights updated in offline batch mode; one complete presentation of the entire training dataset per training epoch (Algorithm 3.4).

Training metric: *MSE* mean-squared error is used for flow and volume-based models. An additional penalty term (sum of square of weights) to regularise the weight values and help to prevent overfitting is used for the depth-based models. The training metric evaluates the error between the ANN output and the target hydrographs during each training epoch. Both metrics

[13] *These apply for all project stages unless explicitly stated.*

evaluate mean squared error; regularisation additionally penalises high magnitudes of weight (approaching or above 1 or -1), tending to lead to lower mean values of weights and hence to a reduced probability of overfitting. This is found to be beneficial to the results obtained for depth, i.e. where hydrographs are of a more continuous shape. However, for flood/spill volume (and to some extent, flow) hydrographs are often less continuous, spending long periods of time at zero or base flow levels, then rapidly and briefly peaking. For these, the MSE metric is found to obtain better results.

Early stopping: *Early stopping* (Caruana et al., 2001; Rafiq et al., 2001) of training is always used, based on a periodic interruption of training to evaluate the partly trained ANN's performance on a single validation event (randomly selected and removed from the set of training events). The training process is interrupted in this way every 50 epochs. In the event that validation error is found to be increasing by more than 1% above the minimum previously measured, training is stopped. This is standard practice to help avoid overfitting.

Maximum training epochs: figures of *1000, 2000, 2500* and *5000* are used for different runs, in order to limit the amount of time taken by the training process, in the event that optimisation stagnates. In practice, relatively few training processes are stopped by this mechanism; early-stopping occurs instead.

Training goal: *0.001:* This is a mechanism provided by the MATLAB NN toolbox, which allows training to be terminated on reducing a training metric error to a specific value. The value is set sufficiently low (0.001) that it is only rarely reached during the number of epochs of the training run. This is so as to attain the best possible performance.

ANN output clamping: this is optionally provided, if appropriate, by means of a post-processing wrapper function for the ANN output hydrographs, as follows:

- Clamping of volume values to a minimum of zero (negative volumes are treated as spurious, so zero is substituted for any negative value of volume

in the output hydrographs). This is only implemented where target hydrographs are found to have no negative values.

- Clamping of manhole flood depths to a maximum of zero: (positive flood depths are treated as spurious, where the InfoWorks manhole method had been set to "lost"). Using the "lost" method means that target hydrographs are truncated during periods of flooding at a maximum depth of zero)[14].

Performance metrics (e.g. Nash Sutcliffe Efficiency Coefficient or R^2) are found to improve, when this technique is used on relevant model runs. Therefore it is always used for the real rainfall experiment/stage runs, where appropriate.

Experiments are also carried out in which a separate ANN is instantiated for each sewerage node ("multiANN"). This is the approach generally adopted by other researchers reported in the literature. Single node models are found to perform marginally better, but at the cost of training time increases by an order of magnitude (each ANN needs training sequentially). The innovation of use of multi-output ANNs for modelling multiple sewerage nodes simultaneously is also worth investigation. Therefore the single node approach is not adopted for the final trials reported for this project.

3.6.2 Training and testing using the RAPIDS ANN tool

The training and testing methodology used for the Design Rainfall and Real Rainfall Experiment/Stages for all three case study catchments is now described. The methodology variation for the Sensitivity Analysis stage is described separately in section 3.10.2.

3.6.2.1 RAPIDS ANN Algorithm

The RAPIDS ANN platform (Duncan et al., 2011, 2013a, 2013b) is used throughout this case study. This implements the moving time windowed approach using lagged ANN inputs indicated in Figure 3.8, which also forms the core of later versions' operation. Input and configuration data are automatically

[14] *Where flooding occurs, this is modelled separately by a flood volume model.*

read from MS Excel files. All ANN output and target hydrographs are written to the output files for evaluation using the agreed set of metrics described below in section 3.6.2.3. The data flow diagram for RAPIDS 1.6 is shown in Figure 3.9.

Figure 3.8 is a graphical representation of the algorithm (Algorithm 3.4) that implements the training and testing regime used in the case study.

Algorithm 3.4: Urban flood prediction: SCG-based ANN training and test

Input: N_{in}: Vector of numbers of lagged timesteps in moving input time window to be used;
N_{hu}: Vector of numbers of hidden units in ANN architecture to be used;
N_{pta}: Vector of numbers of prediction timestep advances to be modelled;
t: timestep; N_t : Vector of numbers of timesteps per event;
R: Rainfall hyetographs for E events [intensity + cumulative for each event];
I_{opt}: Optional additional input variables time-series data {NAPI | Pump states | Tidal levels} for E events;
n_{sig}*: number of input signals in {R,* I_{opt}*}*
U_E: Vector of uses for E events {Trg | Val | Tst};
T: InfoWorks CS-generated target flood hydrograph dataset for a given catchment (section 0) for E events and N_{out} sewer nodes;

Output: N_{in} *x* N_{hu} *x* N_{pta} *x* $E(U_E$*=="Test")* sets of N_{out} ANN model output predicted hydrographs *(*Y_{te}*)* stored in HydroMAT files;

1. *For each* n_{pta} *in* N_{pta} *(prediction timestep advance to be modelled):*
2. **Begin**
3. Shift T by n_{pta} timesteps from origin
4. *Construct training, validation and test datasets:*
5. $X_{tr} \leftarrow \{R, I_{opt}\}(E(U_E$*=="Trg"))* (concatenate rainfall, opt inputs for all training events)
6. $X_{va} \leftarrow \{R, I_{opt}\}(E(U_E$*=="Val"))* (concatenate rainfall, opt inputs for all validation events)
7. $X_{te} \leftarrow \{R, I_{opt}\}(E(U_E$*=="Tst"))* (concatenate rainfall, opt inputs for all test events)
8. $T_{tr} \leftarrow \{T\}(E(U_E$*=="Trg"))* (concatenate target hydrographs for all training events)
9. $T_{va} \leftarrow \{T\}(E(U_E$*=="Val"))* (concatenate target hydrographs for all validation events)
10. $T_{te} \leftarrow \{T\}(E(U_E$*=="Tst"))* (concatenate target hydrographs for all test events)
11. *For each* n_{hu} *in* N_{hu} *(ANN architecture: number of hidden units):*
    Begin
12. *For each* n_{in} *in* N_{in} *(number of lagged timesteps in moving input time window):*
    Begin
13. *Construct parallelised input training dataset (with moving time window):*
14. xtr(*) ← [null];
15. *For each e in* $E(U_E$*=="Trg") (for each training event)*
    Begin
16. *For* $t \leftarrow n_{in}$ *to* $N_t(e) - n_{pta}$ *(for each useable timestep in this event)*
    Begin
17. *For* $s \leftarrow 1$ *to* n_{sig} *(for each input signal)*
    Begin
    Append values in time window for each signal for this timestep:
18. $xtr(t) \leftarrow \{ xtr(t), X_{tr}(t, s), X_{tr}(t-1, s) \ldots X_{tr}(t-n_{in}+1, s)\};$
19. **End;**
20. **End;**
21. **End;**
    xva(*) ← Repeat steps14 – 21 for validation dataset, using X_{va} and
22. $E(U_E$*=="Val")*

23. *xte(*)* ← Repeat steps14 – 21 for test dataset, using X_{te} and $E(U_E$=="Tst")
24. *Use SCG algorithm to train ANN as follows:*
25. *net*(n_{sig} x n_{in}, n_{hu}, n_{out}) (Instantiate ANN object with appropriate architecture)
26. $net.\{W_1,W_2,B_1,B_2\}^0$ ← *rand* (Uniform randomly initialise weights and biases)
27. *Evaluate fitness of ANN:*
28. **Begin**
29. $Y_{tr}(*) \leftarrow f_{net}(xtr(*),net.\{W_1,W_2,B_1,B_2\}^0)$ (Simulate using training dataset)
30. $\epsilon_{tr} \leftarrow mse(T_{tr}, Y_{tr})$ (Compute MS error between target and output)
31. **End;**
$net.\{W_1,W_2,B_1,B_2\}^1 \leftarrow net.\{W_1,W_2,B_1,B_2\}^0 + f(\partial\epsilon/\partial net.W)$
(Compute gradients and update ANN object weights and biases in direction as
32. per SCG algorithm (Møller, 1993))
33. *Train for up to ep == 5000 epochs using batch-mode offline training*
34. **Begin**
35. Repeat steps 28 – 32 for this epoch
36. *Interrupt for validation every v generations:*
37. **Begin**
38. $Y_{va}(*) \leftarrow f_{net}(xva(*),net.\{W_1,W_2,B_1,B_2\}^{ep})$ (Simulate←validation dataset)
39. $\epsilon_{va} \leftarrow mse(T_{va}, Y_{va})$ (Compute MS error between target and output)
IF (early-stopping criterion met OR ep==1000) record "best" ANN and
40. exit SCG training mode
41. **End;**
42. **End;**
43. *Simulate using test dataset and save results and ANN weights and biases:*
Repeat steps 38 – 39 using test dataset: *xte*, net^{best} and $E(U_E$=="Tst")
(On completion of training, simulate with the trained network using the test data-folds and store responses together with ROC evaluation metrics in
44. HydroMAT format – save Y_{te} and T_{te} for each event *(E)* separately)
$net.\{W_1,W_2,B_1,B_2\}^{best}$; $W_{io} \leftarrow W_1 \cdot W_2$
Store the trained weights and biases and combined pathway strength matrix
45. (W_{io})
46. **End;**
47. **End;**
48. **End;**

Note: Evaluations of model performance for each hydrograph using metrics: NSEC, PBIAS, E_{ap}, E_{tp}, NRMSD, M_{C1}, B_{A1}, M_{C2} and BA2; (see section 3.6.2.3) – are computed externally to this algorithm within HydroMAT; optionally also for volume and flow hydrographs, metric evaluation: E_{TV}

Figure 3.8 shows that the rainfall data is first used in hydrodynamic simulations performed by InfoWorks CS, to generate the set of target hydrographs that the RAPIDS ANN is expected to output, once trained. The use of simulation is a substitute for having direct observation data of water levels, volumes and/or flows from the drainage network to be modelled. The simulated

events are divided into 3 sets, to be used for training, validation (during training) and test (following training).

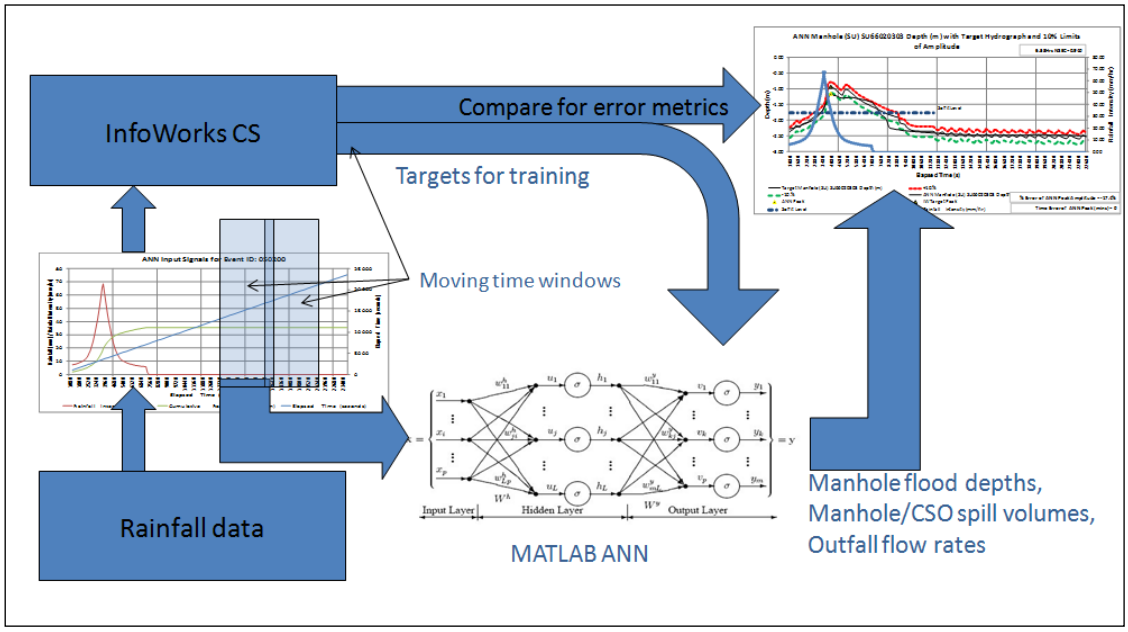


Figure 3.8. Architecture of RAPIDS ANN System to Model and Predict Urban Flood Hydrographs

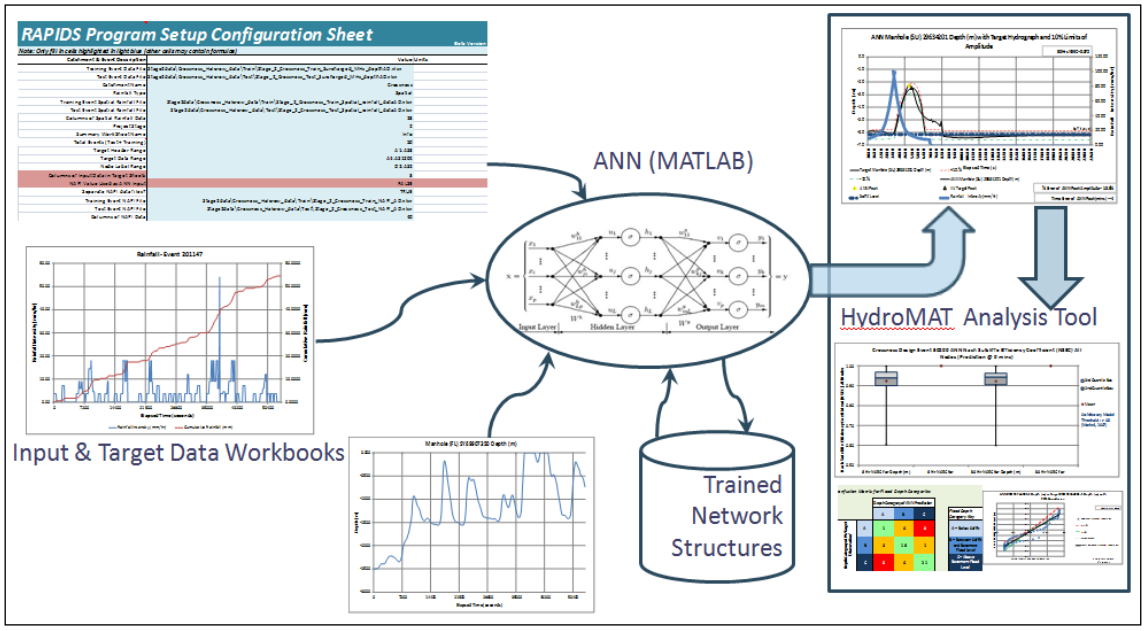


Figure 3.9. Data Flow Diagram for RAPIDS ANN Program

The training events are concatenated and the rainfall (and other optional input data) for these events are applied to the ANN in offline batch mode, whilst evaluating the mean-squared error (MSE) between the ANN output responses and the simulated target hydrographs for the same training events. The SCG algorithm is used to adjust the ANN's weights and biases to minimise the MSE overall. During training, the validation data set is periodically presented to the

ANN and validation MSE is evaluated. Early stopping occurs if validation MSE begins to increase, to prevent overfitting.

Once training is complete, the test data set events are presented one at a time to the ANN and its responses are recorded along with the corresponding target hydrographs. The HydroMAT tool (Figure 3.9) then performs analysis, by comparing the ANN outputs with the targets, to produce the set of metrics described below.

3.6.2.2 Model performance evaluation: HydroMAT Hydrographic Model Analysis Tool

HydroMAT is an MS Excel-based tool for analysis and presentation of ANN regression model performance in graphical and tabular form. RAPIDS1.6 outputs its results directly into HydroMAT as shown in the data flow diagram of Figure 3.9. This illustrates the overall data flow, showing the self-documenting configuration files, automated saving of trained ANN network structures and automated writing of results to HydroMAT. Analysis of results is provided by sewerage node type as well as by measurement unit type. Only results for nodes with non-zero target hydrographs are reported.

3.6.2.3 Model performance evaluation: Metrics implemented by HydroMAT

In order to provide an analysis tool and set of metrics that could be consistently applied across all catchments and consultancy teams involved in the UKWIR RTM project, the HydroMAT (Hydrographical Model Analysis Tool) MS Excel based tool has been developed by the author. All metrics are automatically calculated by HydroMAT from the target and ANN output hydrographs generated by Algorithm 3.4. Moriasi et al. (2007) provide a review of popular metrics used in “watershed” (catchment) evaluations and metrics from this are included in the list below. Others are included by agreement with all UKWIR project partners. The metrics are as follows:

NSEC – Nash Sutcliffe Efficiency Coefficient

This metric evaluates a (“predicted”) time-series (for example an ANN output hydrograph) with respect to a reference target time-series (referred to as

"observed")(for example a hydrograph from InfoWorks CS); and looks at the summed square of the error between the predicted series and the observed series on a sample-by-sample basis. This is normalised by the summed squared difference between each sample of the target hydrograph and the mean of the target hydrograph. The measure is then subtracted from 1 so that NSEC values run between [-∞, +1] with 1 being a perfect prediction; 0 implying that the model only predicts the mean of the observed and negative numbers indicating very poor model performance. For the UKWIR study, NSEC values over 0.85 are taken as "good"; with those over 0.5 as "acceptable". NSEC is implemented by equation (3.1).

NSEC suffers from the problem of becoming increasingly sensitive to differences between predicted and observed hydrographs when there are small amplitudes for target (observed) hydrographs; so any such nodes are removed from the summary analysis. This is reasonable, since such nodes are not likely to be flooding.

$$NSEC = 1 - \left[\frac{\sum_{i=1}^{n} \left(Y_i^{obs} - Y_i^{pred} \right)^2}{\sum_{i=1}^{n} \left(Y_i^{obs} - \bar{Y}^{obs} \right)^2} \right] \qquad (3.1)$$

where: *i* is the index of samples in the time-series; *n* is the total number of samples; Y_i^{obs} is the value of the i^{th} sample of the observed target time-series; Y_i^{pred} is the value of the i^{th} sample of the predicted (ANN) time-series; $\bar{Y}^{obs}$ is the mean value of the observed time-series.

PBIAS – Percentage bias

This metric evaluates a ("predicted") time-series (for example an ANN output hydrograph) with respect to a reference target time-series (referred to as "observed")(for example a hydrograph from InfoWorks CS); and measures the average tendency of the simulated data to be larger or smaller than their observed counterparts. Positive values indicate model underestimation bias, and negative values indicate model overestimation bias. PBIAS is implemented by equation (3.2) and is expressed as a percentage of the sum of the observed

samples over the time series (usually in our case the duration of a rainfall event).

$$PBIAS = \left[\frac{\sum_{i=1}^{n}\left(Y_i^{obs} - Y_i^{pred}\right) * 100}{\sum_{i=1}^{n}\left(Y_i^{obs}\right)}\right] \qquad (3.2)$$

where: *i* is the index of samples in the time-series; *n* is the total number of samples; Y_i^{obs} is the value of the i^{th} sample of the observed target time-series; Y_i^{pred} is the value of the i^{th} sample of the predicted (ANN) time-series.

E_{ap} – Amplitude error of hydrograph peak

E_{ap} and E_{tp} quantify the errors associated only with the hydrograph peak, since this is the most critical period during a rainfall event, when the most significant impacts may occur. Figure 3.10 illustrates these errors, showing the (InfoWorks) observed hydrograph (thin black line) and ANN predicted hydrograph (thick black line).

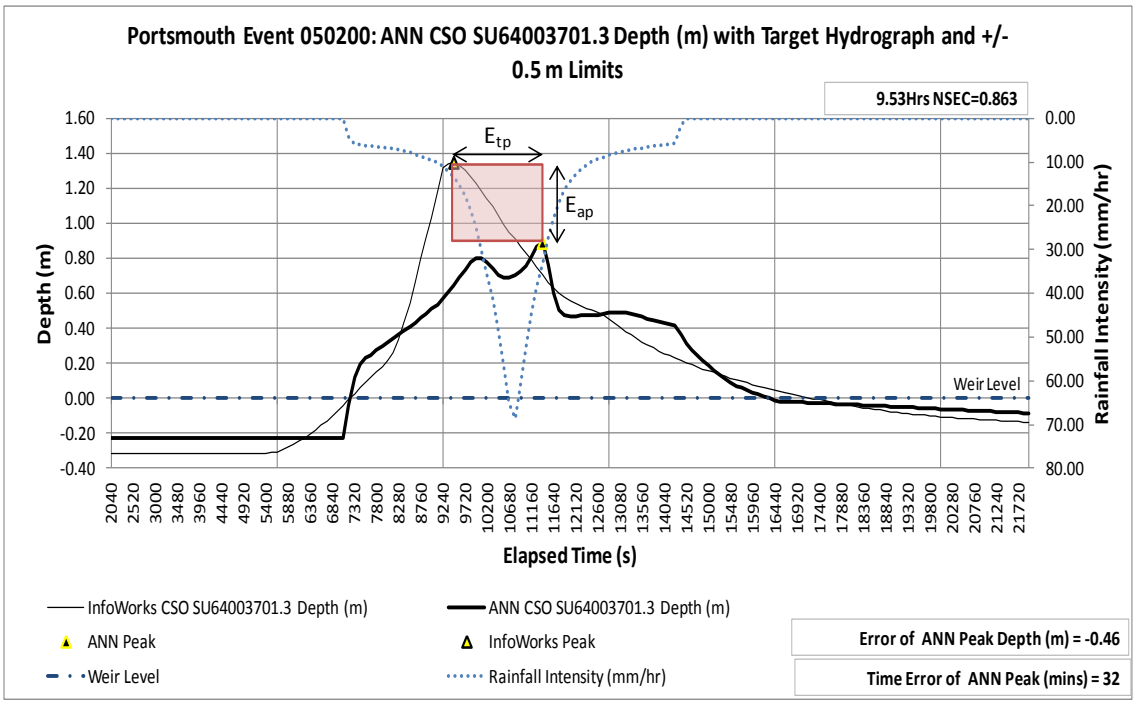


Figure 3.10. Chart of observed and predicted hydrographs showing peak errors

E_{ap} – is defined as amplitude error of hydrograph peak is the height of the pink box in Figure 3.10. This is measured in the units associated with the hydrograph: depths in metres; flow rates in m^3s^{-1}; volumes in cubic metres per

timestep (m^3t^{-1}). This is felt to be more meaningful than converting to a percentage. Equation (3.3) defines:

$$E_{ap} = \left(A_{pk}^{pred} - A_{pk}^{obs}\right) \quad (3.3)$$

where: A_{pk}^{pred} is the amplitude of the peak of the predicted (ANN) hydrograph; A_{pk}^{obs} is the amplitude of the peak of the observed (target) hydrograph. Thus a positive value indicates the peak is over-predicted; whereas a negative value indicates under-prediction.

E_{tp} – Timing error of hydrograph peak

E_{tp} – is defined as timing error of hydrograph peak is the width of the pink box in Figure 3.10. This is measured in minutes. Equation (3.4) defines:

$$E_{tp} = \left(t_{pk}^{pred} - t_{pk}^{obs}\right) \quad (3.4)$$

where: $\mathrm{t}_{\mathrm{pk}}^{\mathrm{pred}}$ is the time of the peak of the predicted (ANN) hydrograph; $\mathrm{t}_{\mathrm{pk}}^{\mathrm{obs}}$ is the time of the peak of the observed (target) hydrograph. Times are measured in elapsed minutes since the start of the rainfall event. Thus a positive value indicates the predicted peak is delayed with respect to the observed; whereas a negative value indicates the predicted peak is advanced with respect to the observed.

NRMSD – Normalised root means square deviation (percentage)

This metric also evaluates a ("predicted") time-series and measures the root mean squared error of the predicted data with respect to their observed counterparts, expressed as a percentage of the observed. Values of NRMSD are always positive, so works well as a pair of metrics in combination with PBIAS. NRMSD is implemented by equation (3.5) and is expressed as a percentage of the peak-peak amplitude of the observed time series.

$$NRMSD = \frac{\sqrt{\frac{\sum_{i=1}^{n}\left(Y_i^{pred} - Y_i^{obs}\right)^2}{n}}}{\left(\max(Y^{obs}) - \min(Y^{obs})\right)} * 100\ \% \qquad (3.5)$$

where: i is the index of samples in the time-series; n is the total number of samples; Y_i^{obs} is the value of the i^{th} sample of the observed target time-series; Y_i^{pred} is the value of the i^{th} sample of the predicted (ANN) time-series.

M_{C1} – Confusion matrix for peak flood depth categories (below cover)

The confusion matrix for peak flood depth categories (below cover) (M_{C1}) is implemented by a post-processor wrapper function on the ANN output "depth" (i.e. water level) hydrographs for manhole sewer nodes. Only the peaks of the predicted and "observed" hydrographs are evaluated in this measure. The schema used implements 3 classes: "flood depth categories" labelled [A | B | C]. The key at the right-hand side of Figure 3.11 defines the categories. These may be more clearly interpreted by referring to the manhole diagram of Figure 3.6.

The matrix provides analysis of the hydrograph peaks of all the ANN output units (equivalent to sewer manholes) being modelled by the multi-output ANN. In the example of Figure 3.11, twenty-three sewer nodes are modelled.

		Depth Category of ANN Prediction		
		A	B	C
Depth Category of InfoWorks 'Observation'	A	0	0	0
	B	0	1	0
	C	0	5	17

Flood Depth Category Key
A = Below Soffit
B = Between Soffit and Basement Flood Level
C = Above Basement Flood Level

Figure 3.11. M_{C1} confusion matrix for flood depth categories (below cover)

The three rows of the matrix correspond with the three depth categories of the observed hydrograph peaks and the three columns correspond with the ANN predicted hydrograph peaks. Numbers indicate the number of nodes

falling into each paired combination of depth categories. In each pair, the ordering is <*predicted*><*observed*>. There are three “pass” combinations: [AA | BB | CC] shaded in green. There are four “caution” combinations, where the predicted and observed categories differ by just one class: [AB | BC | CB | BA] shaded in amber and there are two “fail” combinations, where the predicted and observed categories differ by two classes: [AC | CA] shaded in red. In the illustrated example, there is 1 node in combination BB, 17 nodes in CC and 5 nodes in BC. These last are under-predictions by 1 class, so would fall into the “caution” band.

B_{A1}– Accuracy band for peak flood depth categories (below cover)

B_{A1}– Accuracy band for peak flood depth categories (below cover) is a summary of the confusion matrix *M_{C1}* (above), in which the numbers for the combinations in the “pass” (3), “caution” (4) and “fail” (2) bands are each summed. The example shown in Figure 3.12 corresponds with the confusion matrix in Figure 3.11. There are:

- 0 (AA) + 1 (BB) + 17 (CC) = 18 pass nodes;
- 0 (AB) + 5 (BC) + 0 (CB) + 0 (BA) = 5 caution nodes;
- 0 (AC) + 0 (CA) = 0 fail nodes;

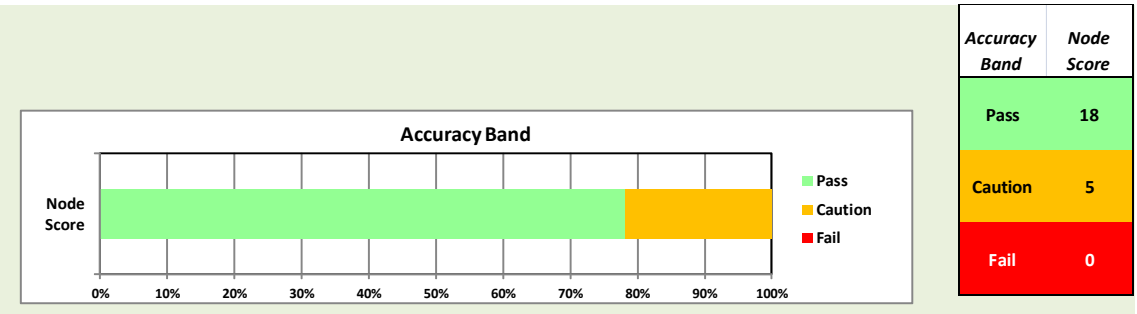


Figure 3.12. Accuracy band for peak flood depth categories (below cover)

The bar graph also shows this as a percentage of nodes falling into each of the 3 bands; here approximately 78% pass and 22% are caution band.

M_{C2} – Confusion matrix for peak flooding (above cover)

The confusion matrix for peak flooding (above cover) (*M_{C2}*) is implemented by a post-processor wrapper function on the ANN output “depth” (i.e. water

level) hydrographs for manhole sewer nodes. Only the peaks of the predicted and “observed” hydrographs are evaluated in this measure. The schema used implements 2 classes: labelled [Flood | No Flood]. Even though the InfoWorks CS manhole flooding mode is set to “lost”, meaning that all “observed” flood levels are clamped not to exceed zero (manhole cover level), the level of zero is treated as “Flood”; meanwhile levels below zero are treated as “No Flood”.

The M_{C2} flood class matrix provides analysis of the hydrograph peaks of all the ANN output units (equivalent to sewer manholes) being modelled by the multi-output ANN. In the example of Figure 3.13, twenty-three sewer nodes are modelled[15].

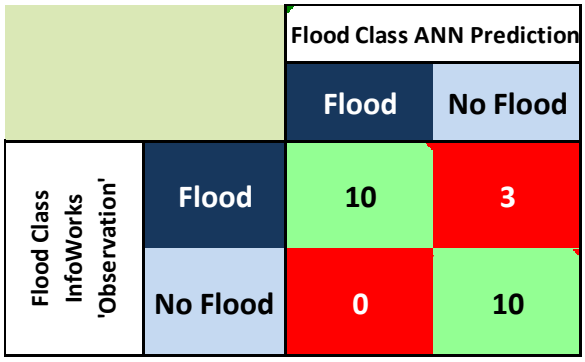


Figure 3.13. M_{C2} - Confusion matrix for peak flooding (above cover)

The two rows of the matrix correspond with the two classes [Flood | No Flood] of the observed hydrograph peaks and the two columns correspond with the ANN predicted hydrograph peaks. Numbers indicate the number of nodes falling into each paired combination of flood classes. In each pair, the ordering is <*predicted*><*observed*>. There are two “pass” combinations: [Flood-Flood | No flood-No flood] shaded in green. Here, “flood” is treated as “positive” and “no flood” as negative. Therefore the “pass” combinations respectively correspond with true positives (TP) and true negatives (TN). There are also two “fail” combinations: [Flood-No Flood | No flood-Flood] shaded in red. These correspond with false positives (FP) and false negatives (FN). In the illustrated example, there are 10 TP nodes, 10 TN nodes, 3 FN nodes (under-predictions) and 0 FP nodes (over-predictions).

[15] *This is the same example as above; flooding nodes are included in depth category C*

B_{A2} – Accuracy band for peak flooding (above cover)

B_{A2} – Accuracy band for peak flooding (above cover) is a summary of the confusion matrix M_{C2} (above), in which the numbers for the combinations in the "pass" (2) and "fail" (2) bands are each summed. The example shown in Figure 3.14 corresponds with the confusion matrix in Figure 3.13. There are:

- 10 (Flood-Flood)(TP) + 10 (No flood-No flood)(TN) = 20 pass nodes;
- 0 (Flood-No flood)(FP) + 3 (No flood-Flood)(FN) = 3 fail nodes;

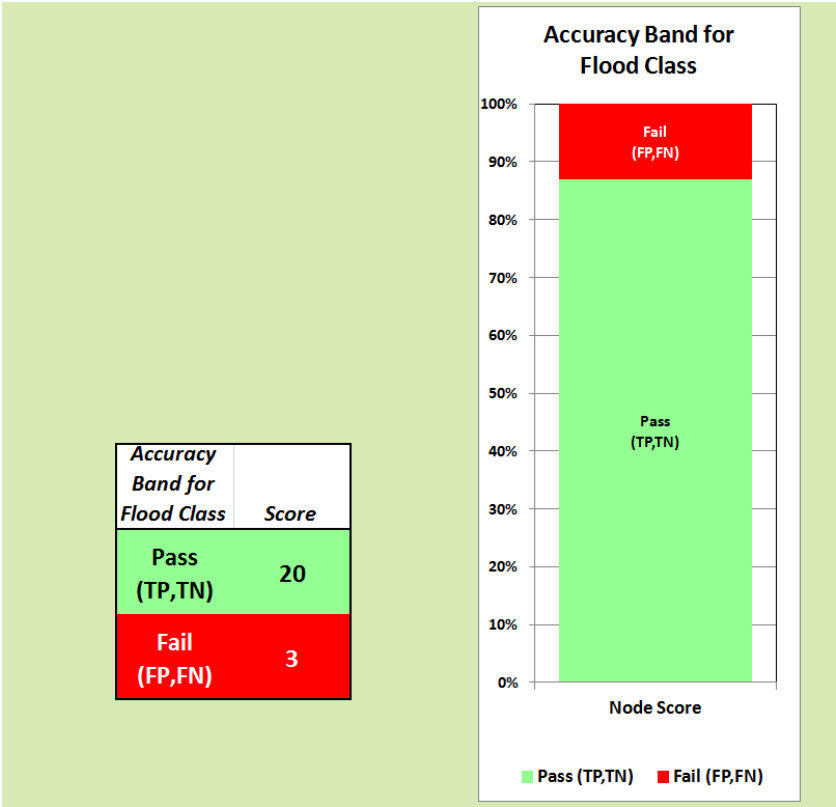


Figure 3.14. Accuracy band for peak flooding (above cover)

The bar graph also converts this to a percentage of nodes falling into each of the 2 bands; here approximately 87% pass and 13% fail.

E_{TV} – Total Volume Error over a rainfall event

E_{TV} – is the Total Volume Error between the ANN output and the target hydrograph over the duration of a rainfall event. This is measured in cubic metres (m^3). This measure only applies to nodes where the measurement type is volume (m^3 per timestep) or flow rate (m^3s^{-1}). Equation (3.6) defines:

$$E_{TV} = \sum_{i=1}^{n} \left[\left(Y_i^{pred} - Y_i^{obs} \right) \times \begin{cases} 1\ (volume\ node) \\ t\ (flow\ rate\ node) \end{cases} \right] \qquad (3.6)$$

where: *i* is the index of samples in the time-series; *n* is the total number of samples for the rainfall event; Y_i^{obs} is the value of the i^{th} sample of the observed target time-series; Y_i^{pred} is the value of the i^{th} sample of the predicted (ANN) time-series; *t* is the number of seconds per timestep[16].

3.6.2.4 Data preparation (design rainfall stage)

The hydrographical time-series datasets used in this study comprise samples at a regular time interval or timestep. In the absence of true observation data from sewer networks, the hydrographs are generated by InfoWorks CS based on the same rainfall hyetographs as are used for the inputs to the ANN models. The InfoWorks output is treated as "ground-truth" for the purposes of this study. They are also referred to as “observed data”.

Table 3.4. Matrix of design rainfall events for Portsmouth

Event No	Event Type	Return Period	Duration	Event Use	Event ID
	Design / Real	rrr (Years)	d.dd (Hours)	Trg / Tst	Format rrrddd
1	Design	1	0.5	Trg	001050
2	Design	1	1	Tst	001100
3	Design	1	2	Trg	001200
4	Design	1	4	Trg	001400
5	Design	5	0.5	Trg	005050
6	Design	5	1	Trg	005100
7	Design	5	2	Tst	005200
8	Design	5	4	Trg	005400
9	Design	20	0.5	Trg	020050
10	Design	20	1	Tst	020100
11	Design	20	2	Trg	020200
12	Design	20	4	Trg	020400
13	Design	50	0.5	Trg	050050
14	Design	50	1	Trg	050100
15	Design	50	2	Tst	050200
16	Design	50	4	Trg	050400

In the Design Rainfall Experiment/Stage of this case study, input data consistently uses 16 design rainfall events (Pilgrim, 2001; Wilson, 1990), with

[16] *Flow rate nodes are subject to an error since the formula assumes the flow rate remains constant for the duration of each timestep and the measured value is only an instantaneous sample for 1 second per timestep.*

12 being used for training and 4 for test evaluation of performance of the trained networks. There are 4 durations of rainfall event: [0.5 | 1 | 2 | 4] hours and 4 return periods: [1 | 5 | 20 | 50] years. All time-series for the events are buffered to be of the same duration by varying the run-off periods following each event. An example for the Portsmouth catchment is shown in Table 3.4.

Throughout the case study, Dorchester and Portsmouth uses a 2-minute timestep, whereas Crossness uses 5-minutes. These are the simulation timesteps used in InfoWorks.

3.6.2.5 Data preparation and modelling strategy – based on early results

During early trials, a single ANN model is initially used to predict hydrograph outputs for all sewer node types. This is found to generate unsatisfactory results. Further trials reveal that there are a number of reasons for this:

- Combining all hydrograph measurement unit types and node types together means that amplitudes differ greatly, resulting in relatively poor performance for nodes with small amplitude hydrographs.

- Cumulative hydrographs (e.g. flood volumes) are not suitable to be modelled directly by the ANN. Instead, volume hydrographs are modelled in units of m^3 per timestep. If required, cumulative volume hydrographs could easily be produced from these by a post-processing wrapper function.

- Differing hydrograph shapes (e.g. flood volumes compared with manhole depths) are not readily modelled by a single ANN. Instead, better results are obtained when a separate ANN model is created for each sewerage node type and measurement unit type.

As a result of this, the results reported here follow the guidelines of modelling different node types and measurement unit types with different ANN models and modelling using quantities per timestep, rather than cumulative quantities.

3.6.2.6 Data preparation (real rainfall stage)

Case study real rainfall experiment/stage data preparation is carried out as follows:

Each catchment provides records for 50 real rainfall events together with the (InfoWorks CS) “observed” response hydrographs at the sewer nodes to be modelled. For all catchments, 5 of the rainfall events are reserved to use for test, following completion of training, a single event is used for validation during training and the other 44 events are used for the training process itself.

When selecting the rainfall events to use for test, it is important to ensure that they lie within the envelope of cumulative rainfall for those events used for the model training. In the example for the Dorchester catchment, shown in Figure 3.15, four of the test events [201112, 201129, 201132 and 201147] comply with this requirement; whereas event 201126 has a period between 5 and 8 hours after the start of the event, where the cumulative rainfall runs very close to the upper edge of the envelope described by the training events. This is likely to yield poor results for that event. In the event that no real rainfall events exist of sufficient intensity it is standard practice to augment real rainfall profiles either by a fixed factor or by using a stochastic rainfall generator to produce artificially intense events (Cameron et al., 1999).

The ANN software is then run; first to train and then to test the models. Evaluation of each test event automatically produces a HydroMAT file containing results for all the agreed metrics (section 3.6.2.3) for manual inspection and analysis and collation of summary results across all events and sewerage nodes.

Table 3.5 summarises the ANN models developed for each of the catchments for the Real Rainfall Experiment/Stage. A selection of typical results is presented in the results section for concision, rather than attempting a comprehensive and detailed listing.

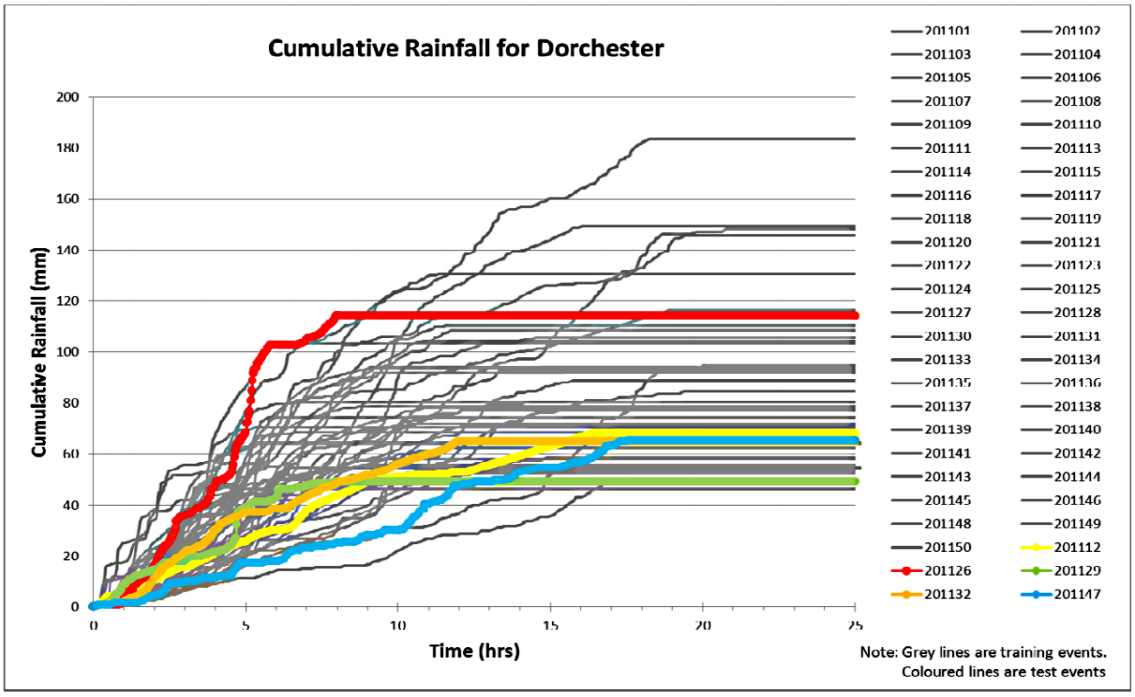


Figure 3.15. Dorchester cumulative rainfall profiles for 45 training and 5 test events (UKWIR, 2012)

Table 3.5. ANN models developed for case study catchments – Real Rainfall Stage

Measurement type	Crossness	Dorchester	Portsmouth
CSO Depth	ANN C1	ANN D1	ANN P1
Flood manhole Depth	ANN C2	ANN D2	ANN P2
Surcharge manhole Depth	ANN C3	ANN D2	ANN P2
CSO Spill Volume	ANN C4	ANN D3	ANN P3
Flood manhole Flood Volume	ANN C5	ANN D3	ANN P3
In-sewer Flow rate	ANN C6	ANN D4	-
CSO Flow rate	ANN C7	-	-

In the case of Dorchester, parallel models are built with and without NAPI as an optional input to assess whether this improves capabilities of the ANN tool to predict an incident. Figure 3.16 illustrates a typical set of inputs for a real rainfall event for the Dorchester catchment. These are shown prior to normalisation and parallelisation for the moving lagged input time window.

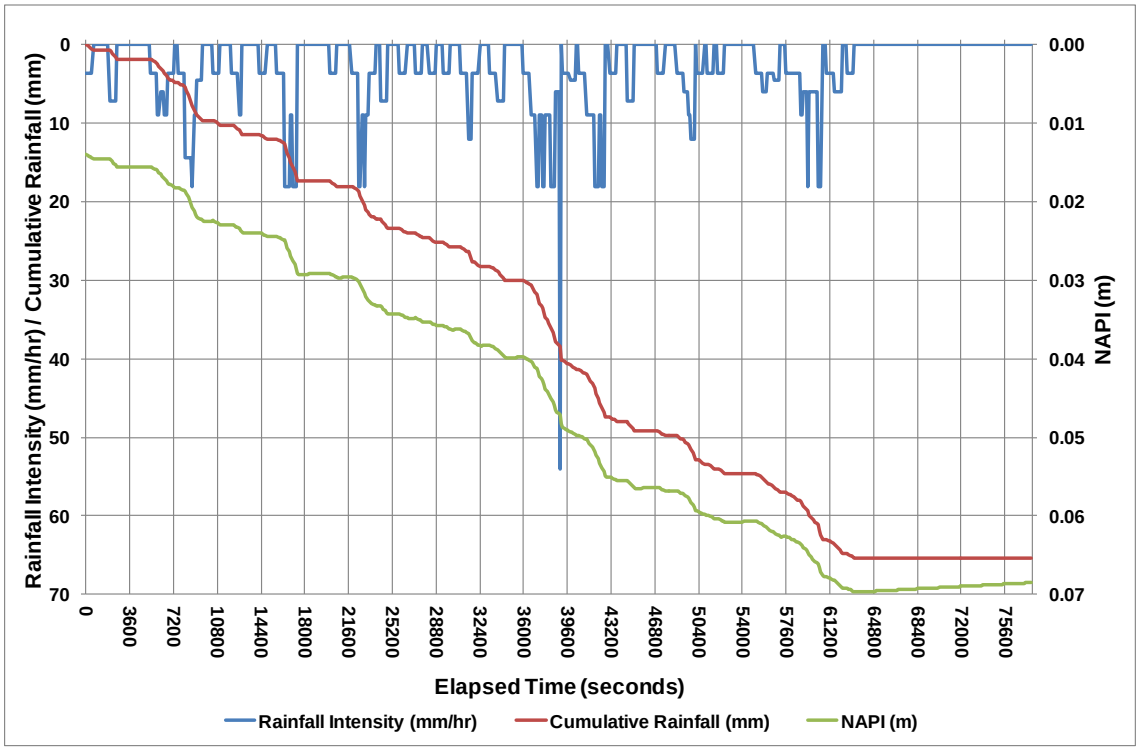


Figure 3.16. Signals applied to an ANN model for one of the Real Rainfall Stage test events (201147) before normalisation - Dorchester

The corresponding training event InfoWorks hydrographs, for the selected sewer node locations to be modelled, are used as target signals for training each ANN output. Masking is used at the event boundaries to prevent predictions based on input data from each previous event being confounded with target data from each next event. The width of the mask at the start of an event is the same as the lagged moving input time window width and the width of the mask at the end of an event is the same as the prediction timestep advance.

3.7 Performance Results (Design Rainfall Experiment/Stage)

Results presented in this section are for the three catchments, using design rainfall events as described in section 3.6.2.4. First, examples of typical ANN model output hydrographs compared with the InfoWorks target hydrographs are presented:

3.7.1 Individual node hydrographs

Figure 3.17 presents ANN output (thick black line) and corresponding InfoWorks target (thin black line) water depth hydrographs for a single

surcharging manhole (SY68900501) for a 50-year return period (RP), 2-hour duration design rainfall event in the catchment of Dorchester. It shows the Nash-Sutcliffe score of 0.85 over the 6-hour duration of the event and run-off period, so represents an example of a hydrograph on the borderline between the defined NS level for “good” and “acceptable”. Also shown (red and green dashed lines) are agreed +/-20% instantaneous limits on the target hydrograph. It can be seen that the ANN output remains within these limits for the majority of the time. Also shown is the error of peak amplitude of 0.6%, which corresponds with approximately 10mm in this case. Timing error of peak is +6 minutes (delay). The (design) rainfall hyetograph is also charted on the secondary axis. In the case of this manhole, the soffit level is shown (blue-dot-dash line) as being 1.55m (approx) below cover.

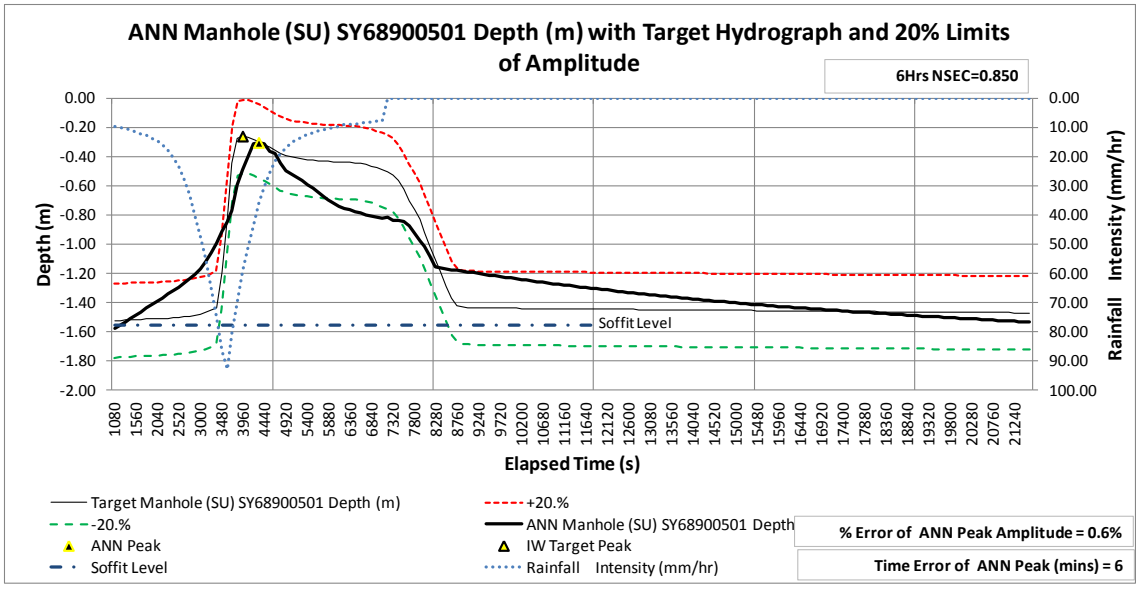


Figure 3.17. Dorchester design rainfall event 50-year RP / 2-hour duration (manhole depth)

Figure 3.18 displays a CSO spill flow rate hydrograph within the Crossness catchment, showing the tendency for the flow to be at the zero level for long periods and exhibit a sudden peak, making modelling more challenging than in the case of water depth. The NS score for this node, however, is “good” at 0.91. The design rainfall event is for a 20-year return period and 1-hour duration. Also clearly displayed is the ability of the ANN to synchronise the peak of its output with that of the target hydrograph, despite these being delayed with respect to the peak of the rainfall hyetograph. The ANN is able to do this

through the use of the time-lagged inputs within the moving input time window. This is modelled using ANN architecture with 100 hidden units.

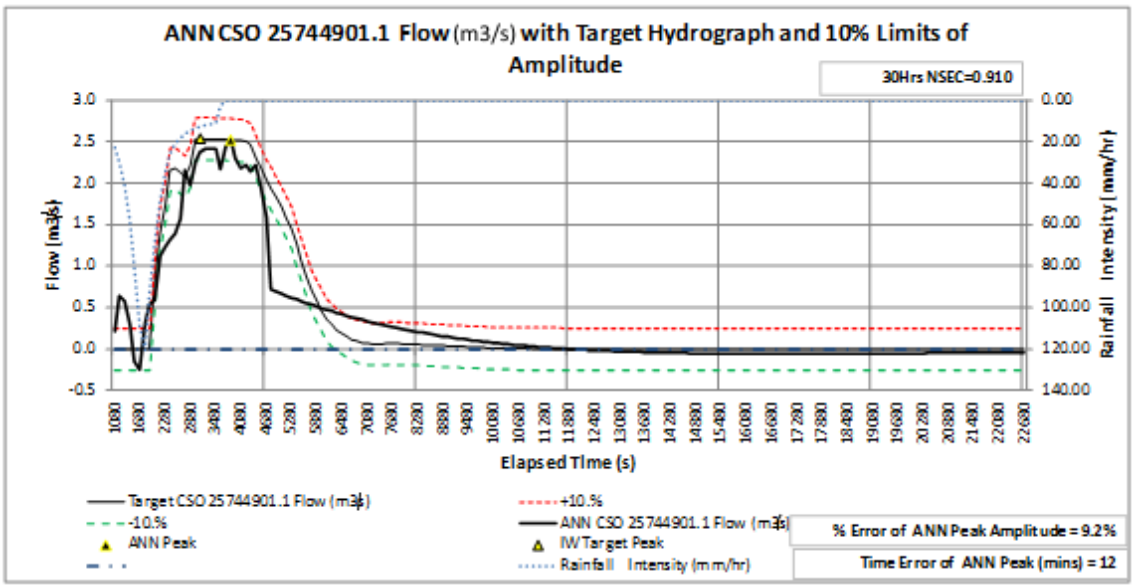


Figure 3.18. Crossness design rainfall event 20-year RP / 1-hour duration (CSO flow rate).

Figure 3.19 shows a hydrograph of flood volume from the cover of a manhole in the Portsmouth drainage network for a 5-year return period and 2-hour duration design rainfall event. The target hydrograph exhibits extremely rapid onset and cessation of flow. The ANN (with 10-hidden units in this case) does not manage to model such rapid transitions and its response is nearer to the shape of the input rainfall hyetograph, incorporating a delay. This tends therefore to suggest under-fitting. The architecture might benefit from having more hidden units. Nonetheless the NS of 0.8 is still “acceptable”.

These 3 hydrographs are presented mainly as typical examples. During the course of the case study, thousands of hydrographs have been produced and inspected and it is impossible to document them all here.

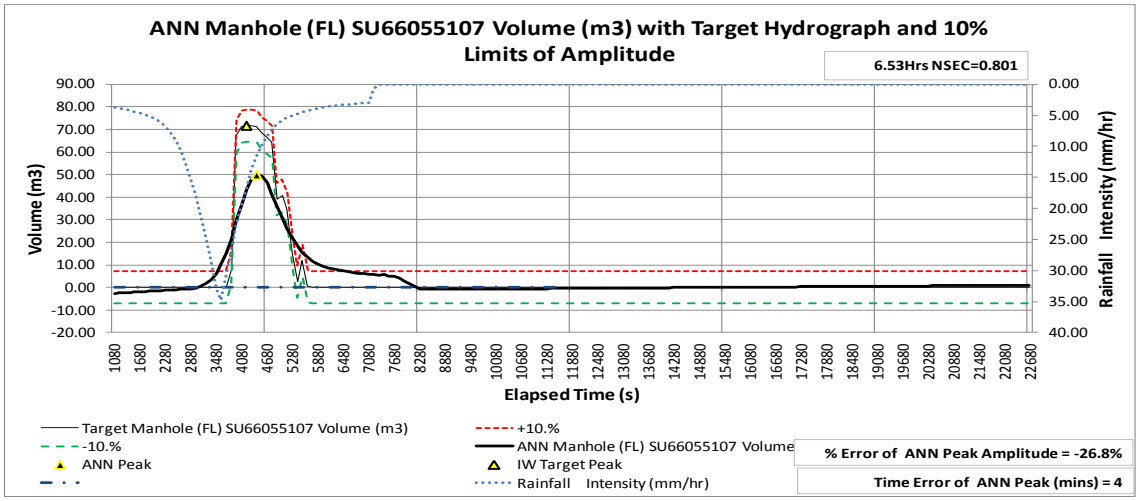


Figure 3.19. Portsmouth design rainfall event 5-year RP / 2-hour duration (manhole volume)

3.7.2 ANN training regime for Crossness volume models

This section summarises the training of multi-output ANNs modelling several sewer nodes for flood volume. The ANN has an input time window of 10 x 5-minute timesteps for each input signal, giving a lag of between 0 and 45-minutes. There are three input signals (rainfall intensity, cumulative rainfall and elapsed time since start of each event), so a total of 30-input features. There are 100 hidden units and 23 output units – one for each manhole being modelled. The training regime is Scaled Conjugate Gradients (SCG) for 2500 epochs, offline batch mode. Training error (MSE) progress during the training run is displayed in Figure 3.20. This is summed across the 23 outputs.

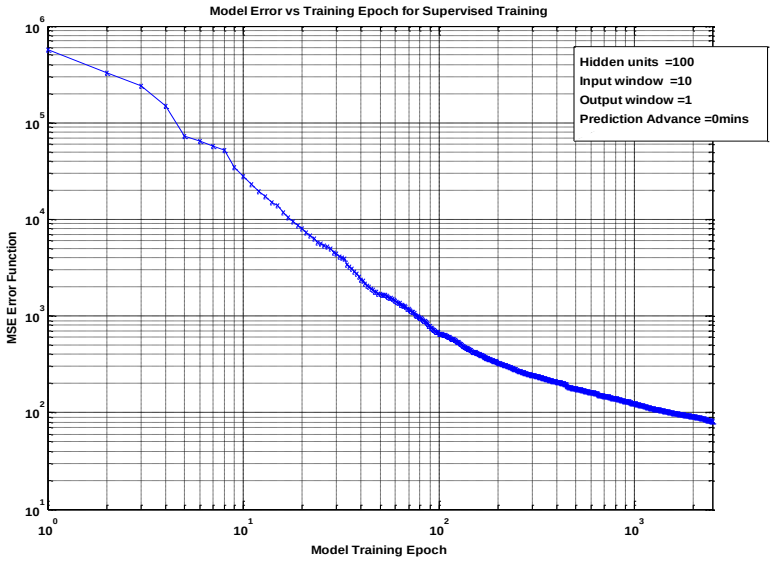


Figure 3.20. ANN MSE training error progress over 2500 epochs

As can be seen from the plot of validation error in Figure 3.21, this continues to reduce during the whole 2500 epochs, so training could have continued beyond this time, even though early stopping is active. A running average over 5 samples of the validation error is shown on the plot as the solid black line. An increase of this is used for the early stopping criterion.

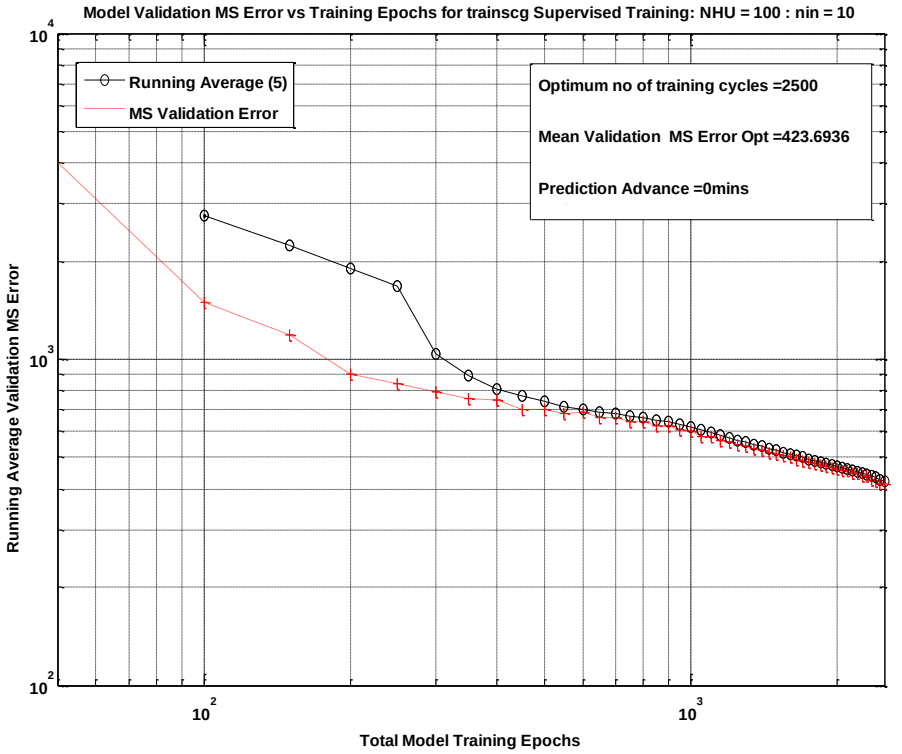


Figure 3.21. ANN MS validation error progress over 2500 epochs

In this case, training is halted by reaching the setting for the maximum number of epochs (2500). This is reasonable as the network is quite a large one, with 5423 parameters.

3.7.3 Summaries of NS scores across all output units of each ANN

This section summarises some of the NSEC results from groups of units, being the outputs of multi-output ANNs modelling several nodes in the given drainage network.

Figure 3.22 shows the range of NS scores attained by the set of 23 output units for each of four design rainfall events. The raw hydrographs these are based on are for manhole flood volumes within the Crossness drainage network. These show a considerable spread of results but are to some extent misleading, since manholes with little or no flooding are very likely to produce

very large negative NS scores (due to the low reference amplitude of the target hydrograph).

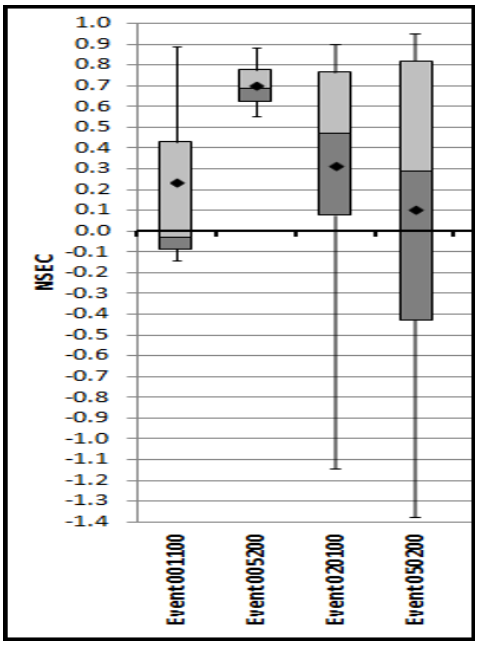


Figure 3.22. Crossness manhole flood volume: NS scores for 4 design rainfall events

Throughout the results section, nodes for which target hydrograph amplitudes are low are excluded from the NSEC metric. The upper limits for these exclusions are shown in

Table 3.6 for the three types of node modelled. Nodes with greater target hydrograph amplitudes than these are included in the analyses of spreads of results for the NSEC metric.

Table 3.6. NSEC metric: exclusion thresholds for different node types

Node Type	Node Exclusion Thresholds	Units
Depth (m)	0.40	m
Flow (m3s)	0.10	m^3s^{-1}
Volume (m3)	3.00	m^3 timestep^{-1}

In Figure 3.22 the events are presented left-to-right having increasing intensity. More nodes are therefore excluded for the events towards the left, due to less flooding occurring. Inspection of the hydrographs shows that performance for event 050200 (50-year RP / 2-hour duration) is overall the best, with the most nodes participating. This problem with low-amplitude “observed” time-series is a consistent feature of the Nash-Sutcliffe Efficiency Coefficient (NSEC) metric in its use with flow rate and volume hydrographs.

Figure 3.23 presents a similar chart for a combined ANN model with 39 outputs, 20 outputs for surcharged manholes and 19 for CSOs. ANN architecture is 10-hidden units and, as above, 30-input features, giving a total of

739 parameters. The ANN predicts flood depths for manholes or spill depths for CSOs. The NSEC scores for this model exhibit a much clearer pattern, with only 10 node exclusions over the two events of lower intensity (001100 and 005200). For each event, the scores for Manhole (SU) and CSO type nodes are displayed in separate boxes. It can be seen that CSO nodes in this model perform significantly better (at the 95% significance level) than their manhole counterparts for the same event, apart from for the most intense event (050200). This result is based on a 1-tailed Student's T-test assuming unequal variance. Inspection of the 050200 event hydrographs for the manhole nodes for which NSEC falls below the threshold for "good" of 0.85 reveals that these are for manholes in the upstream region of the catchment with correspondingly sharp hydrograph peak shapes (typically for both the onset of surcharge and the run-off).

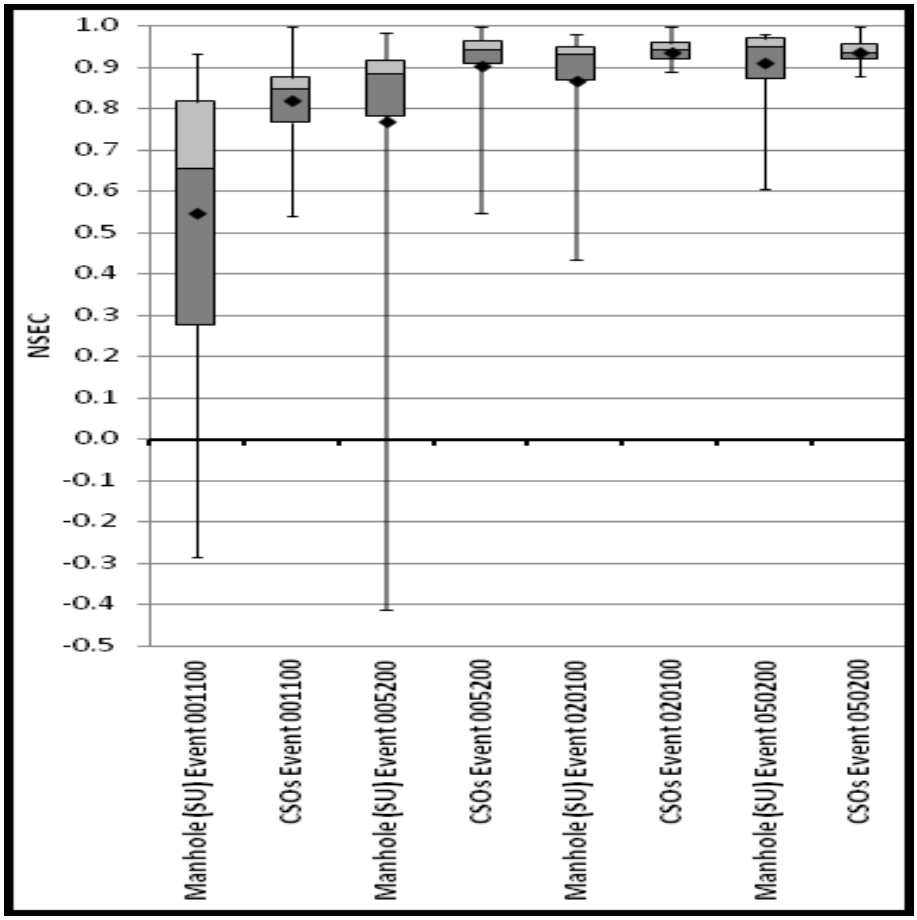


Figure 3.23. Crossness surcharged manhole and CSO depths: NS scores for 4 design rainfall events

Also significant is the pattern of improving NSEC scores for both manholes and CSOs with increasing intensity of event. Again this is as expected with increasing amplitude of the target hydrographs.

Table 3.7. Crossness: depth nodes vs NS classes

NS class	# Nodes
Good	108
Satisfactory	29
Poor	9
Total	**146**

Overall, the majority of NS scores fall into the "good" class. Table 3.7 shows a count of the numbers of nodes in each of the 3 Nash-Sutcliffe score classes (excluding the 10 nodes that are excluded altogether!). It is interesting that this model works as well as it does, given that two node types are represented, despite the fact that the measurement units are the same (Depth (m)) for both node types. This may be due in part to the simplified hydrograph profiles when using design rainfall events with single (Laplace distributed) peak.

Figure 3.24 presents a chart for a combined ANN model again with 39 outputs, 1 outfall link (flow (m^3s^{-1})) output, 19 CSOs (flow (m^3s^{-1})) and 19 CSOs (volume (m3)). Here the emphasis is on a comparison of results for different types of measurement units (flow / volume). The outfall link output is grouped in with the CSO (flow (m^3s^{-1})) nodes. ANN architecture is again 10-hidden units and, as above, 30-input features.

Eight nodes are excluded again from the assessment of the 001100 event, due to zero flow/volume on the target hydrographs. This has skewed the box-and-whisker for volume (m^3) Event 001100 (left hand side of chart) as most of the eight excluded nodes belong in that group.

Two clear patterns emerge:

- That the range of NS scores generally improves with increased intensity of design rainfall event;
- That the range of NS scores for volume nodes is generally higher for the same event than the NS scores for flow nodes. A 1-tailed T-test confirms this with 95% significance level in this instance. This is in part due to the flow rates only being sampled momentarily once per timestep as opposed to the volumes being summed over the whole timestep period.

Table 3.8 gives the number of nodes falling into the NS score classifications of good (>0.85), satisfactory (>0.5) and poor and shows very similar results to the results for the manhole and CSO depth node model.

Table 3.8. Crossness: Flow/volume nodes vs NS classes

NS class	# Nodes
Good	112
Satisfactory	29
Poor	7
Total	**148**

Similar results are also reported for the Dorchester and Portsmouth catchments for case study design rainfall experiment/stage events; so are not presented here in the interests of brevity.

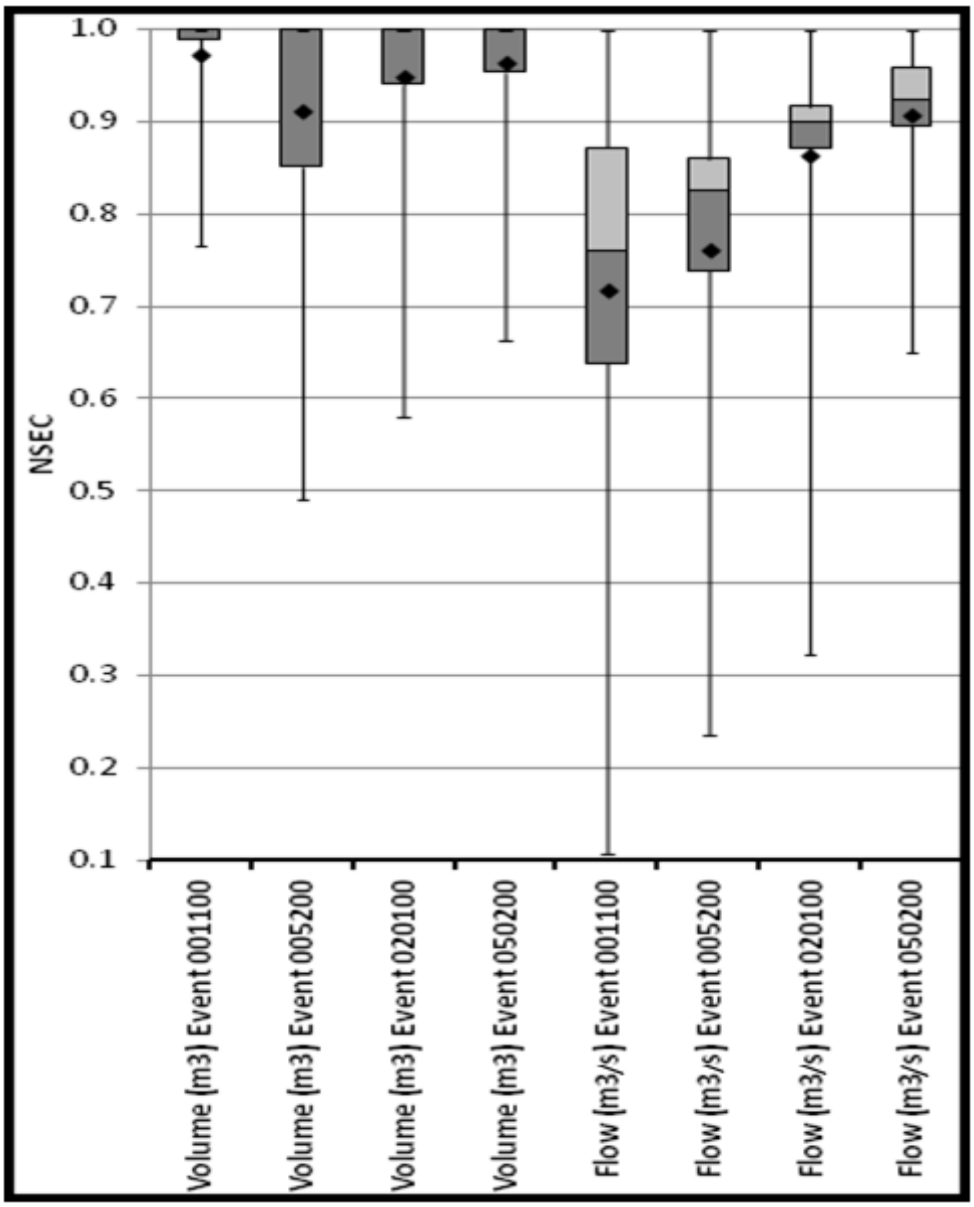


Figure 3.24. Crossness outfall link and CSO flow + volume: NS scores for 4 design rainfall events

Dorchester is used for reporting results for the real rainfall experiment/stage and Portsmouth is used extensively in the Sensitivity Analysis stage for reporting limits on prediction using ANNs with real rainfall input.

3.7.4 Summaries of E_{TV} total volume error across all output units of an ANN

Figure 3.25 presents in graphical form a log-log chart of total volume error (E_{TV}) for the 19 ANN units predicting Flow (m^3/s) shown in the right-hand-most box-and-whisker of Figure 3.24. This is for a 50-year Return Period; 2-hour duration event. These results are the volume errors summed over the entire rainfall event and runoff period duration of 30-hours.

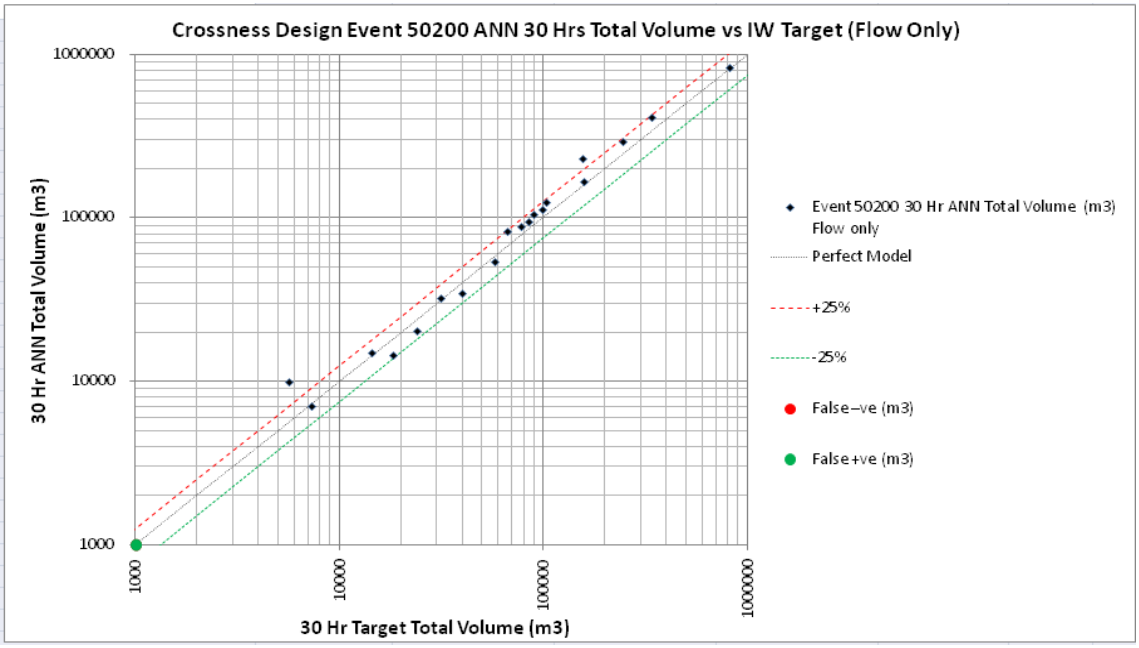


Figure 3.25. Crossness E_{TV} for 19 Flow (m^3/s) nodes

The axes of the chart are “observed” target total volume on the x-axis and ANN predicted total volume on the y-axis.

A logarithmic scale is used because the total volumes for the nodes vary over 3 orders of magnitude. As can be seen, all but two of the 19 nodes remain within the limits (agreed with the project partners as operationally reasonable) of +/- 25% on E_{TV}. The two nodes that are outside these limits have over-predicted volumes by approximately 56% (3,500m^3) and 37% (60,000m^3). Overall, the green dot at the chart’s origin indicates that total over (false positive) or under (false negative) prediction is less than 1,000m^3 summed over all 19 nodes. The volume nodes from the same model perform similarly, so their chart is not reproduced here.

3.7.5 Confusion matrices and accuracy bands across all output units of an ANN

The M_{C1} confusion matrix is now presented in Figure 3.26 for the surcharged manholes predicting water depth presented on the chart of Figure 3.23. In this chart, peak depths for just the 20 manholes (not CSOs) are included. The flood depth category key relates to the manholes only. 18 of the 20 nodes covered are in the “pass” band and 2 manholes are in the caution band (BA), which is an over-prediction. Therefore 90% of nodes are in the pass band, with 10% in the caution band and none in the fail band.

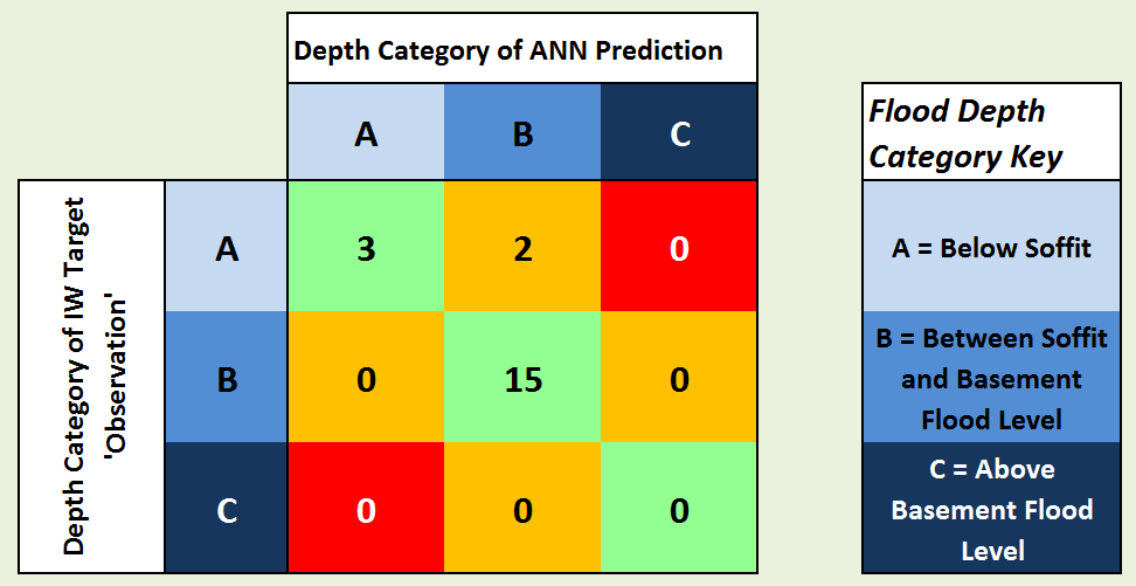


Figure 3.26. Crossness flood depth category confusion matrix M_{C1} for rainfall event 005200

The over-prediction is typical of the pattern that less intense rainfall events tend to be over-predicted, whereas the more intense ones tend to be under-predicted. Event 005200 is a 5-year return period, 2-hour duration design rainfall event, so of relatively low intensity.

Figure 3.27 shows the M_{C2} confusion matrix and corresponding B_{A2} accuracy band for flooding above manhole cover / spills over the CSO weir. This is for the same ANN and rainfall event as in Figure 3.23 and Figure 3.26. This demonstrates 100% classification accuracy in this case, with no false positives or false negatives.

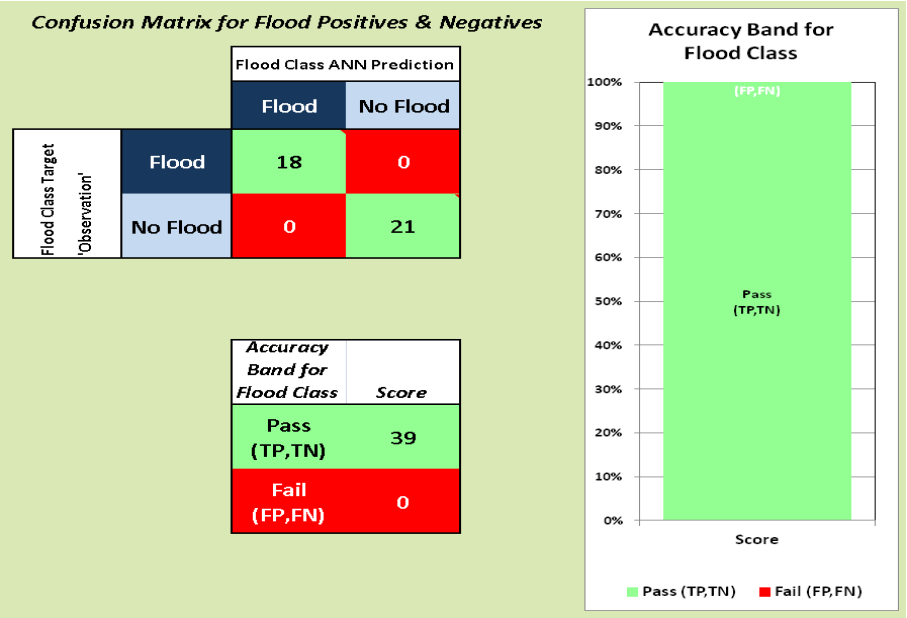


Figure 3.27. Crossness M_{C2} / B_{A2} flood class confusion matrix and accuracy band

3.7.6 Peak amplitude E_{ap} and timing E_{tp} errors across all output units of an ANN

Figure 3.28 shows the spread of values of timing error for the peak of the hydrograph for the same set of 20 outputs for surcharged manholes and 19 for CSOs as used for Figure 3.23. The x-axis presents the results for surcharged manholes on the left 4 box-and-whiskers and those for CSO nodes are on the right hand four. The y-axis is scaled in minutes, with negative values meaning the predicted peak is early and positive values meaning it is delayed. The majority of units can be seen to predict the peak within a few minutes of the target. However, the exception is for CSOs on the 50-year 2-hour duration most intense event. Here all nodes are delayed. As CSOs tend to be in the downstream part of a catchment, their hydrograph profile tends to be more spread out than some of the other nodes, particularly for high intensity events; so it is possible that small errors in amplitude and wave-shape can result in significant displacement of the peak.

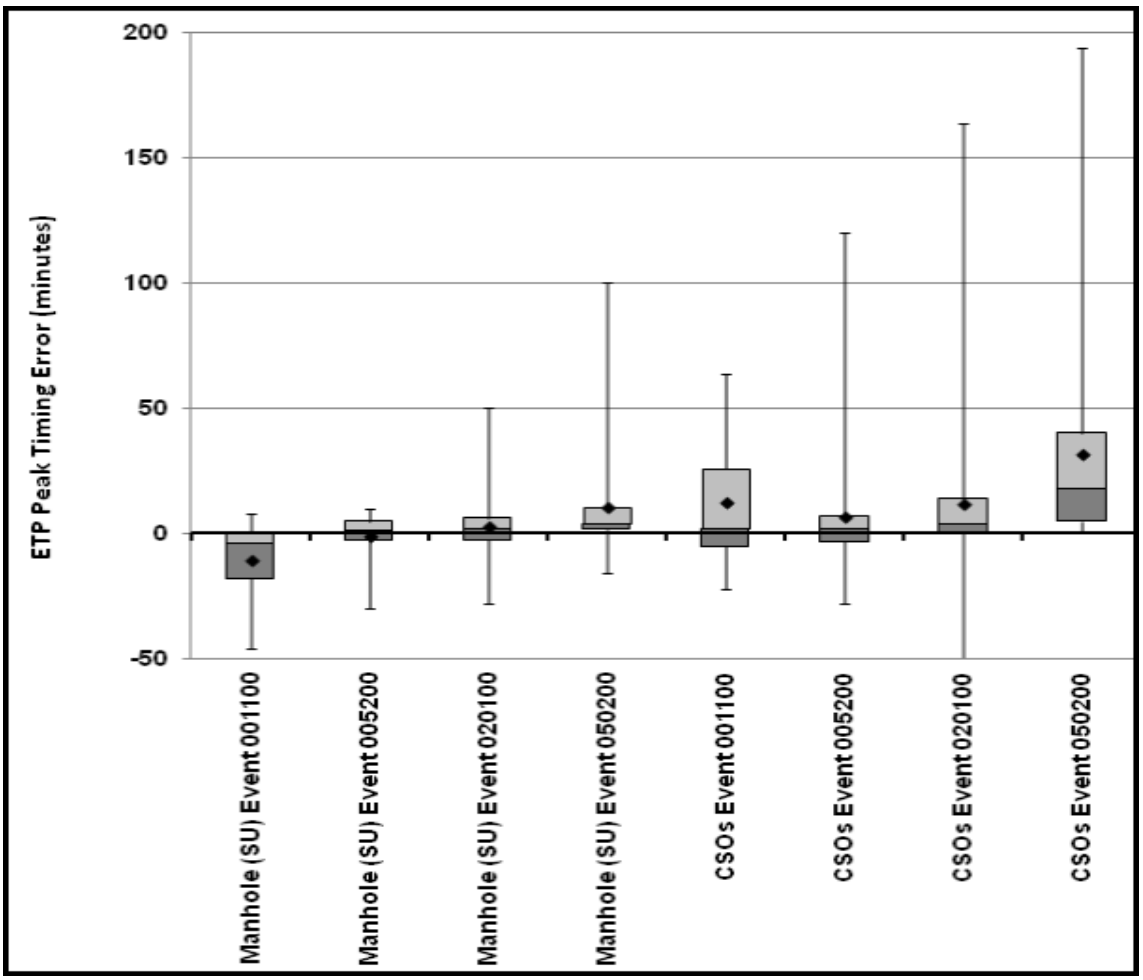


Figure 3.28. Crossness depth nodes E_{TP} peak timing error for 4 design rainfall events

Another feature of Figure 3.28 is the tendency for there to be outliers with (especially) significant delays in the peak. These are usually in a single node per event (out of the 19 or 20). Using the hydrograph display tool in HydroMAT it is possible to drill down into the data and find out the causes behind these outliers. Figure 3.29 illustrates with the example of a hydrograph from a CSO node with a “good” NSEC score of 0.959, yet an error of peak timing of +56 minutes. It can be observed that a small relative amplitude change to the heights of the 2 ANN output peaks could result in the timing error reducing to within a few minutes of the target. Given the variety of wave-shapes that the single multi-unit ANN has to reproduce, it is perhaps not surprising that these timing errors are observed on a few nodes.

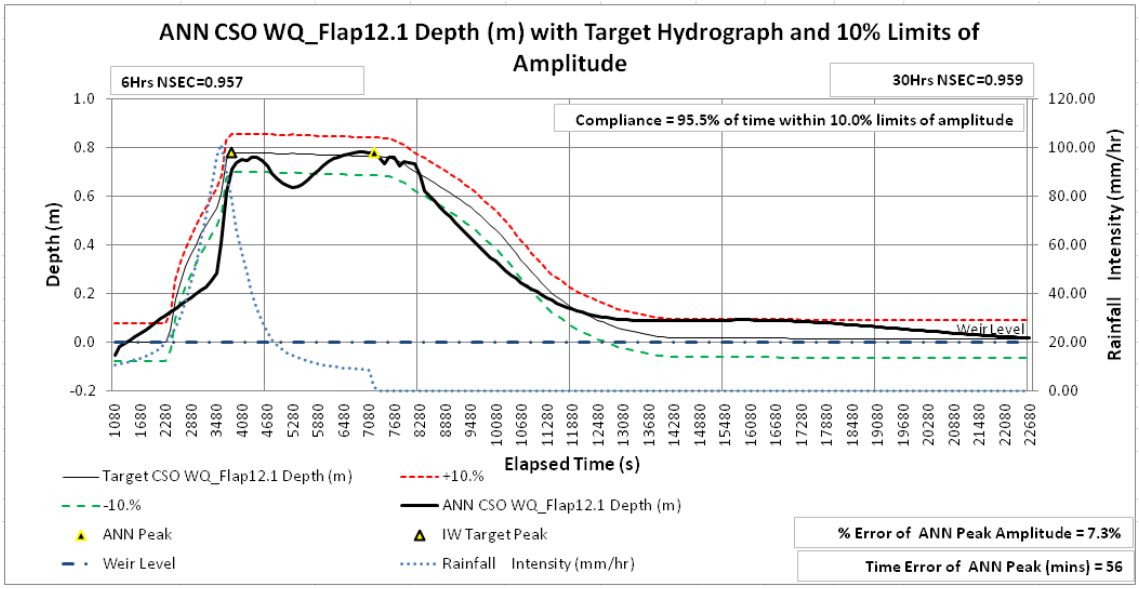


Figure 3.29. Crossness hydrographical example of E_{TP} outlier CSO node

Figure 3.30 provides a summary of spread of values of peak amplitude error. From this it is noticeable that, especially for manholes, the amplitude of the peak is over-predicted for the 2 least intense rainfall events quite significantly. Otherwise prediction accuracy for the peak is within +/-50% in almost every case.

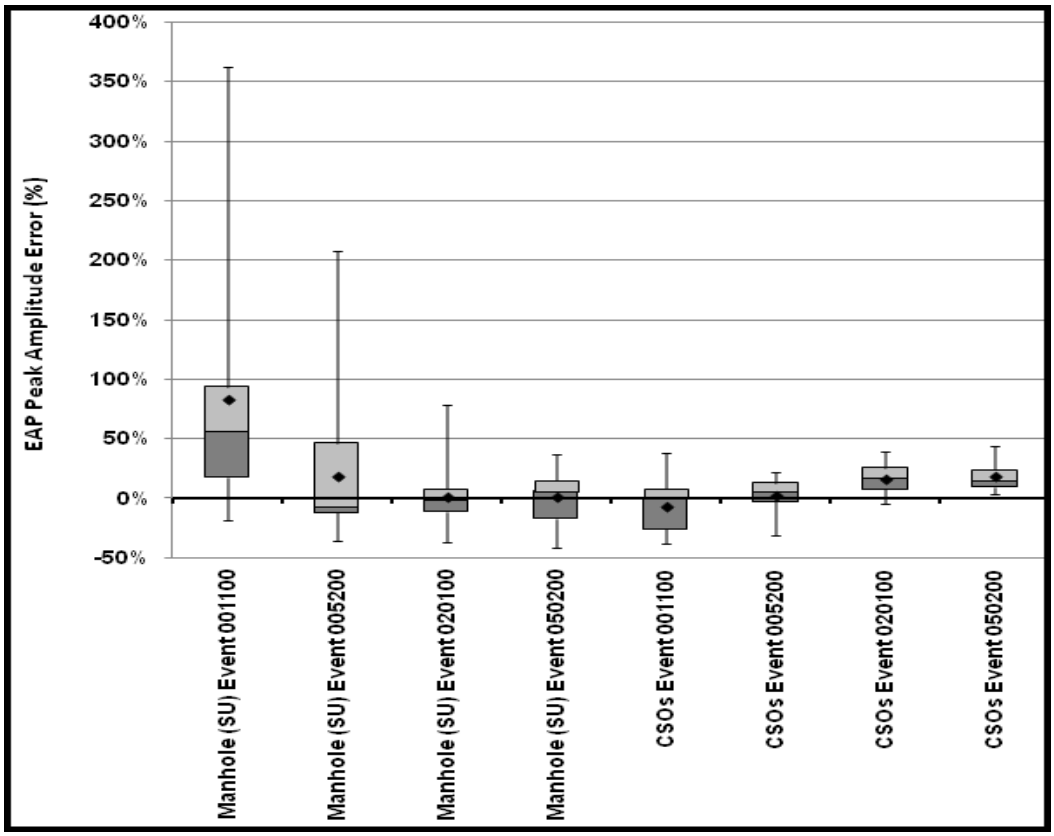


Figure 3.30. Crossness depth nodes E_{AP} peak amplitude error for 4 design rainfall events

The problem of over-prediction of the small event peaks is arguably a less serious problem than under-prediction of the peaks for the large events, when the most impact is likely to occur. The high percentage error for a small amplitude event is a feature of using percentage as the measurement unit. This is effectively rectified in the reporting of the real rainfall experiment/stage results, by using the measurement units of the hydrograph to report the error.

3.7.7 PBIAS – Percentage bias errors across all output units of an ANN

In order to examine the amount of ANN under or over prediction for the 4 design rainfall events, the PBIAS metric is suitable, as shown in Figure 3.31. This shows that the percentage bias of the ANN model output units for the manholes is extremely small; typically less than ±1%. However, for the CSOs the spread of PBIAS values is typically in the range ±10%. In the case of the two most intense rainfall events (020100 and 050200), there is a tendency to over-predict (negative PBIAS). This is due to this being a combined model with both manholes and CSOs, in which the amplitude of the CSO signals are smaller than those for the manholes.

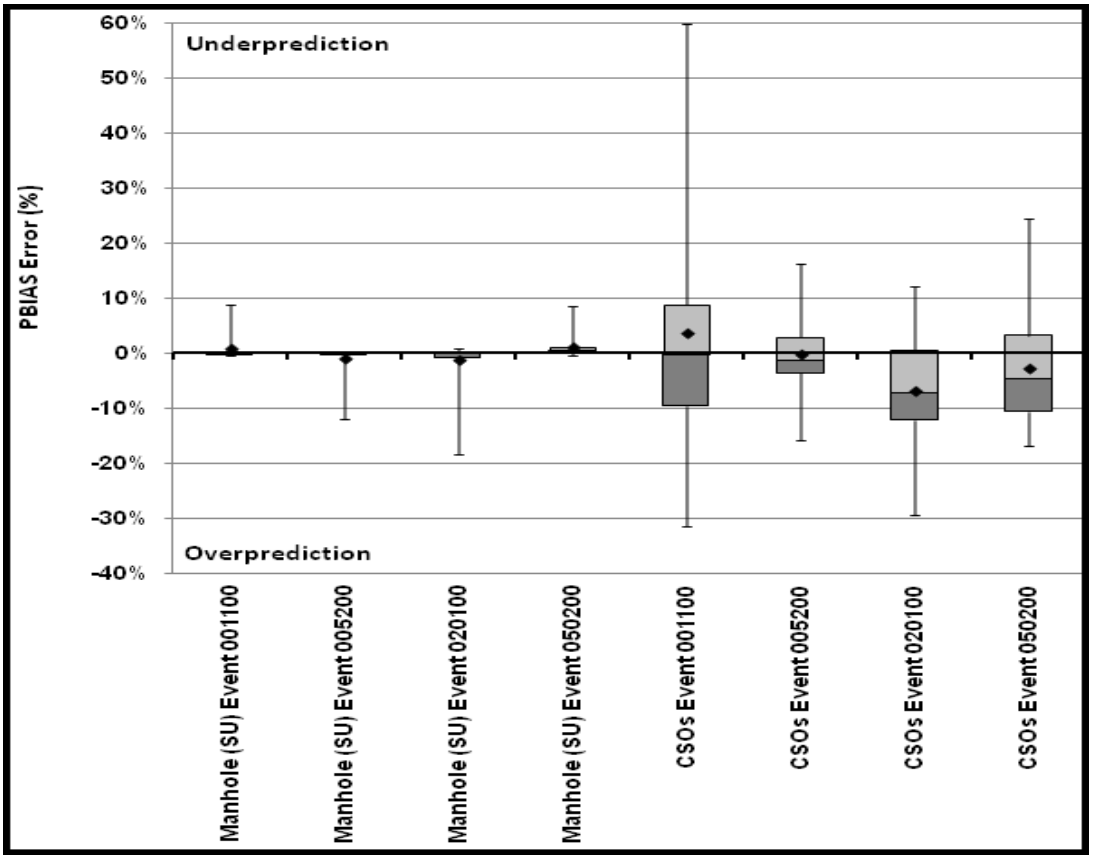


Figure 3.31. Crossness depth nodes PBIAS % error for 4 design rainfall events

3.8 Performance Results (Real Rainfall Experiment/Stage)

Results presented in this section are for the three catchments, using real rainfall events as described in section 3.6.2.6. Initially the results for Dorchester are presented. For this catchment, the spatially-uniform real rainfall intensities are enhanced in order to ensure that some flooding occurs at key nodes during each event. The ANN architecture uses 10 x 2-minute input timesteps, giving time lags between 0 and 18-minutes with either 3 or 4 input signals (rainfall intensity, cumulative rainfall and elapsed time, plus (optionally) NAPI). This totals either 30 or 40 input features. 10 hidden units are used and 40-output units, giving a total of 750 or 820 network parameters. The same training regime as described in section 3.7.2 is used.

3.8.1 Individual node hydrographs

3.8.1.1 Dorchester catchment

One of the objectives of the Dorchester case study is to evaluate the benefit (or otherwise) of using NAPI as model input feature. In this section ANN models with 40-output units predict depth for 20 surcharged and 20 flooding manholes. First, example hydrographs are presented.

Figure 3.32 illustrates a hydrograph for a typical flooding manhole node with NSEC=0.401 for real rainfall event 201147; one of the 5 test events. The ANN model for this chart does not include NAPI as an input. Figure 3.33 is the same node's hydrograph for the ANN model using NAPI as an additional input. This clearly shows little effect from its use in this case; a result that is typical for the flooding manholes. However, there is a very small improvement in NSEC to 0.409 over the complete 14-hour event.

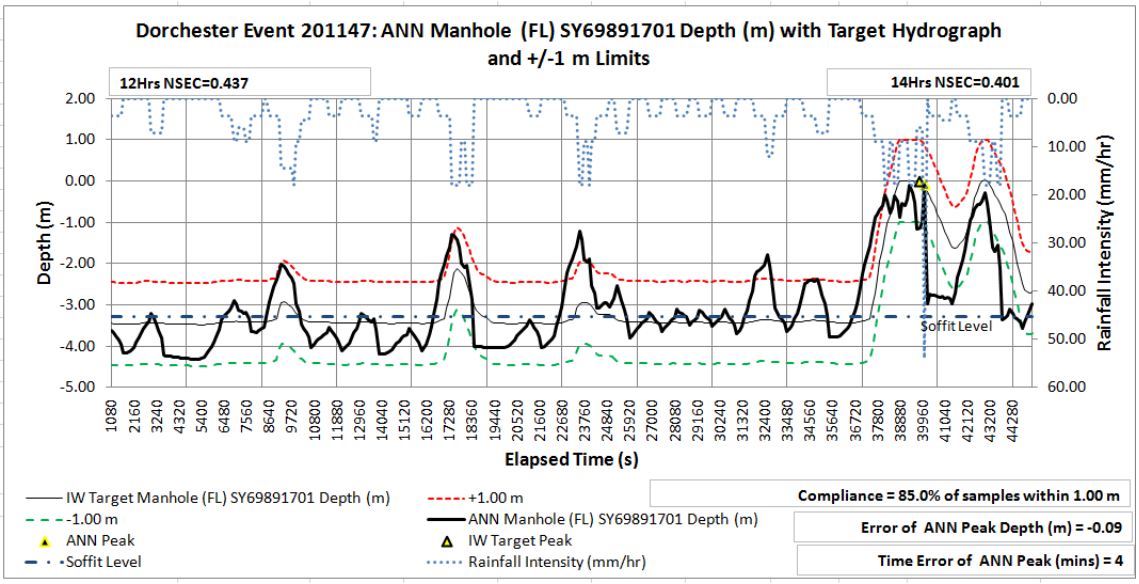


Figure 3.32. Dorchester hydrograph for flooding manhole: real rainfall event 201147

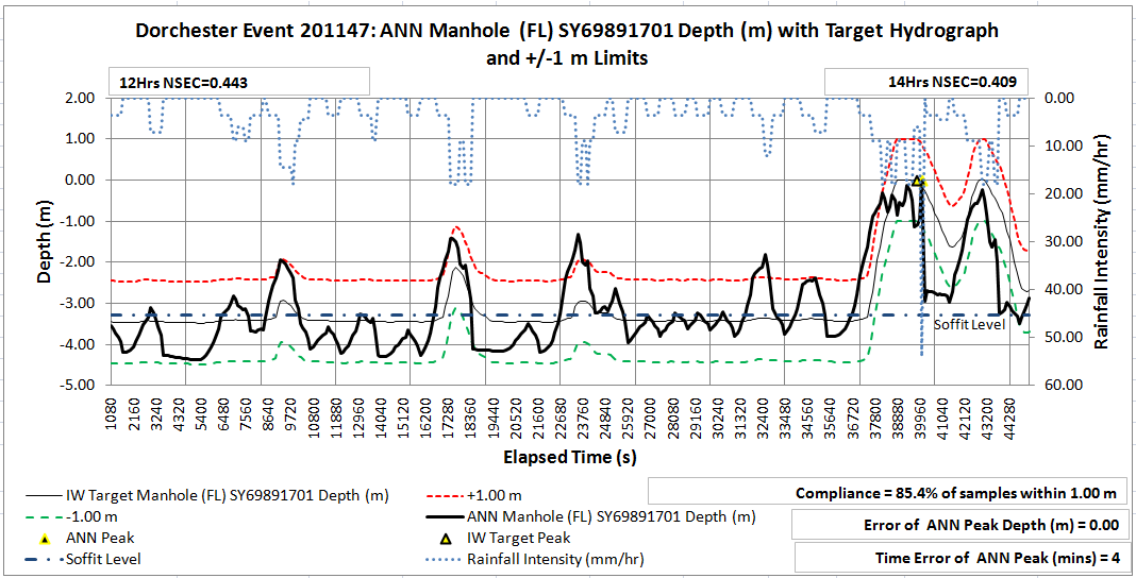


Figure 3.33. Dorchester hydrograph for flooding manhole: real rainfall event 201147 with NAPI input

3.8.1.2 Crossness catchment

The Crossness (South London) catchment is a much more challenging model to build, with 23 raingauges representing spatially varying rainfall and optionally an additional 40 NAPI levels sampled at various points in the catchment. The challenge for the training algorithm is to optimise within a decision space of potentially thousands of dimensions (given the multiplication of inputs within a lagged moving time-window). Nonetheless a model is attempted for spill volumes at 19 CSO nodes. Figure 3.34 illustrates the difficulty the model has with returning to the zero-level following each spill. The NS score for this node and event is -0.37, which is above average for this

model, but very poor. The rainfall shown is for just the first of the 23 raingauges used.

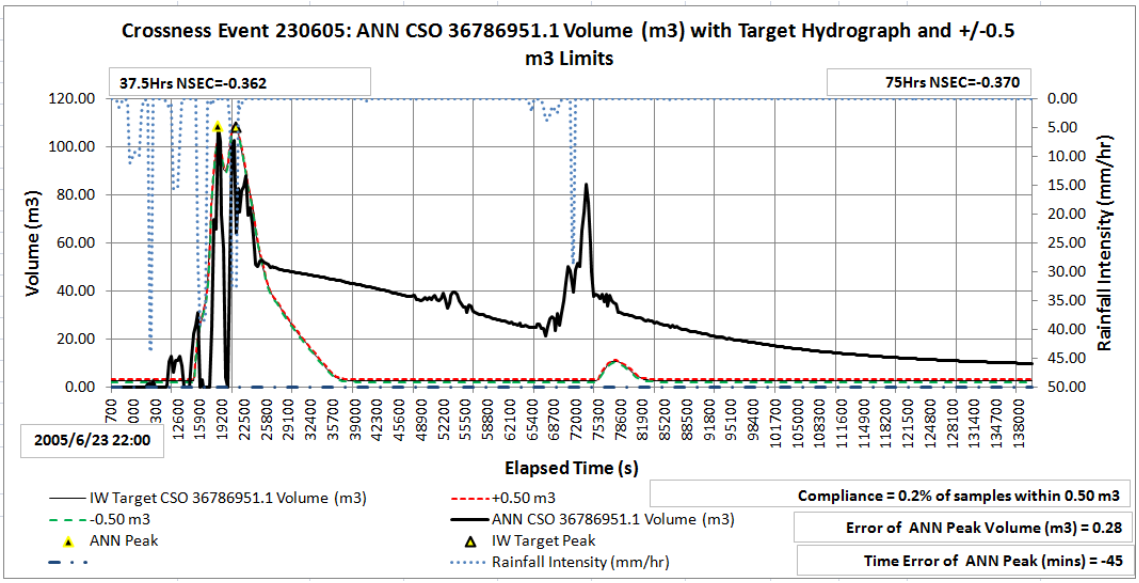


Figure 3.34. Crossness typical hydrograph for CSO spill volume with 23-raingauge spatial rainfall and 40-NAPI inputs (x10 timesteps)

As a result of these (expected) poor results, a number of subsidiary models are built for example sub-sections of the catchment. One such model is for the same single CSO as shown in Figure 3.34, just using the three nearest raingauges as input. The hydrograph for this model for the same event is shown in Figure 3.35. As can be seen, the NS score for this node and model is 0.636, which falls into the “satisfactory” class.

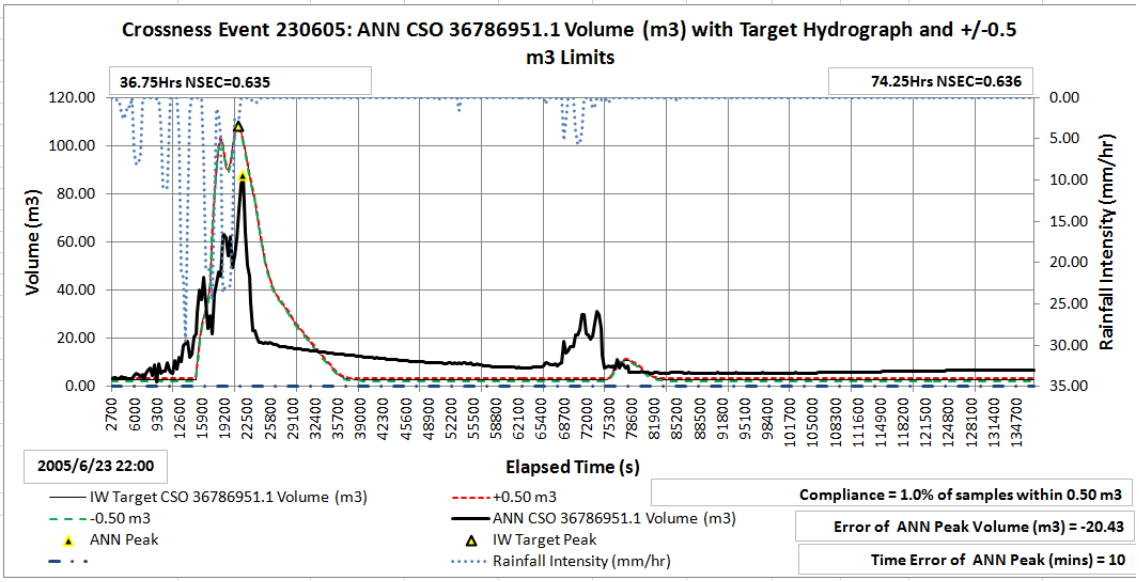


Figure 3.35. Crossness local sub-model hydrograph for CSO spill volume with 3-raingauge spatial rainfall inputs (x10 timesteps)

The difficulties experienced in attempting to model this catchment have led to the search for an automatable methodology for input feature selection and reduction and has resulted in the work described in chapters 4 and 5. It is hoped to be able to extend this to a fully automatable way of building rapid ANN-based flood prediction models for large catchments such as Crossness.

3.8.2 Summaries of NS scores across all output units of each ANN

3.8.2.1 Dorchester catchment

Using the same models as in section 3.8.1, Figure 3.36 shows the spread of NS scores over three of the 5 test rainfall events for the Dorchester catchment. The nodes are grouped into two major groups: surcharged manholes (MS) on the left and flooding manholes (MF) on the right. Each of these groups is sub-divided into 3 pairs (for the 3 events) and within each pair: for an ANN model without NAPI as input (left) and with NAPI as input (right).

As with the design rainfall stage, a number of nodes are excluded from this trial for some or all of the events, due to the target hydrographs being of insufficient amplitude. This is acceptable, because these nodes are not involved in flooding when being excluded. In Figure 3.36, it can be seen that the surcharged manhole group performs better overall than the flooding manhole group. In the case of surcharged manholes, the models with NAPI included as input perform marginally better than their counterparts without NAPI. Conversely, for flooding manholes, the inclusion of NAPI as input appears to make little difference. This is confirmed by the results in Table 3.9, which show Student's 2-tailed, paired T-tests comparing populations of NS scores for nodes where NAPI is used as input with those where it is not. In the case of all 3 rainfall events for surcharged (SU) manholes, the NAPI-included models have higher NS scores at the 95% significance level. However, for flooding manholes (FL), the NS scores are not significantly different, whether or not NAPI is used as an input.

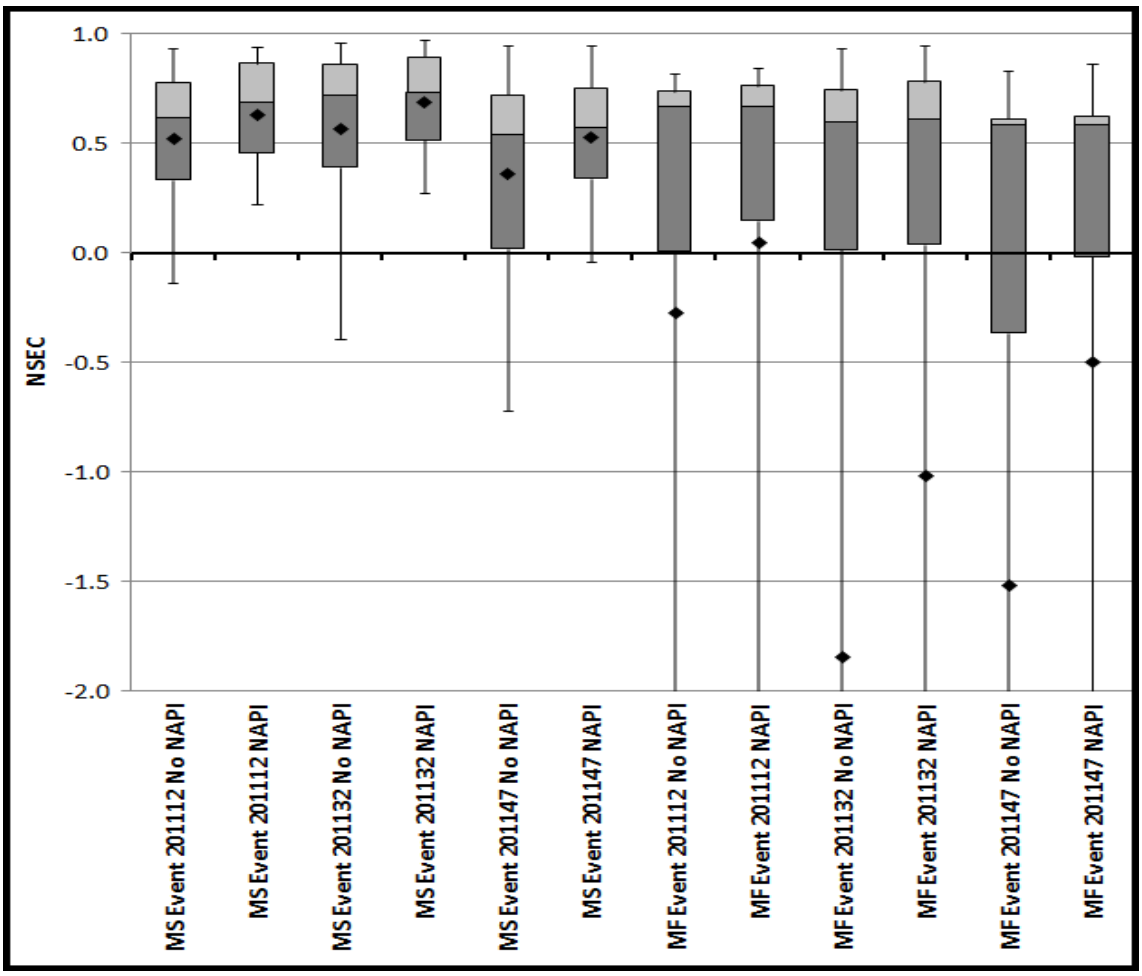


Figure 3.36. Dorchester spread of NSEC values for manhole flood depth with and without NAPI input

Table 3.10. Comparison of ANNs with and without NAPI as input

Node Type	EventID	2-tailed paired T-test
Manhole (SU)	201112	0.008
Manhole (SU)	201132	0.034
Manhole (SU)	201147	0.024
Manhole (FL)	201112	0.061
Manhole (FL)	201132	0.132
Manhole (FL)	201147	0.147

Table 3.9. Summary of NS score classes

NS class	# MS Nodes	# MF Nodes	# Total Nodes
Good	23	7	30
Satisfactory	41	48	89
Poor	32	37	69
Total	**96**	**92**	**188**

Table 3.9 summarises the numbers of nodes overall, for surcharged manholes (MS) and flooding manholes (MF) that fall into the same NS score classes as used in section 3.7.2 and Table 3.8.

Again it is evident from this that the flooding manholes overall perform more poorly than the surcharged manhole nodes. Figure 3.36 shows that for the flooding manhole nodes, there are several nodes with sub-zero NS scores. The chart is truncated at -2 so as to show the detail for the majority of nodes; but some NS scores are less than -15. Examination of the hydrographs for these nodes shows that the ANNs with 10-hidden units (as in this trial) have not been able to follow the variety of hydrograph shapes in every case, given the complexity of the rainfall intensity input signal. There is also a tendency for ANN model hydrographs to respond too much to short-term variations in rainfall, when compared with the “observed” hydrographs.

3.8.2.2 Crossness catchment

Figure 3.37 combines the NS scores for four models:

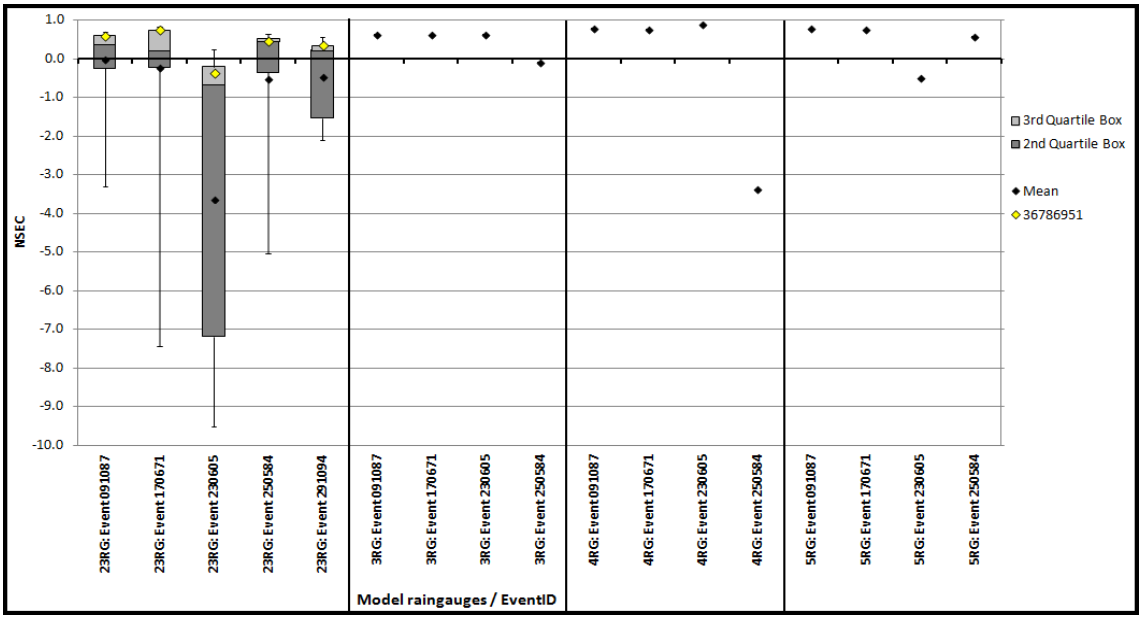


Figure 3.37. Crossness spread of NSEC values for full model and 3 sub-models

- Left section: full model using 23 raingauges modelling spill volumes for 19 CSO’s for 5 rainfall events; results for CSO 36786951 are marked with yellow diamonds for comparison with the other 3 sections
- Second left section: sub-model using 3 raingauges modelling spill volume at a single CSO (36786951) for 4-events
- Second right section: sub-model using 4 raingauges modelling spill volume at a single CSO (36786951) for 4-events

- Right section: sub-model using 5 raingauges modelling spill volume at a single CSO (36786951) for 4-events.

The right-hand 3 sections are modelling a single node. Therefore there are no box-and-whiskers to show spreads of NS scores. The “mean” value markers show the NS scores for the single CSO (36786951) for these models.

Table 3.11 summarises the numbers of nodes for the 4 Crossness models falling into each of the NS score classes. Nodes are counted once for each rainfall event. *nRG* represents the number of raingauges used as inputs in each model. Only the 4-raingauge model managed a “good” NS score on a single event (out of the 4 events included for its model).

Table 3.11. Summary of NS score classes for Crossness full and 3 sub-models (CSO volume)

NS class	23RG # Nodes	3RG # Nodes	4RG # Nodes	5RG # Nodes
Good	0	0	1	0
Satisfactory	19	3	2	3
Poor	45	1	1	1
Total	**64**	**4**	**4**	**4**

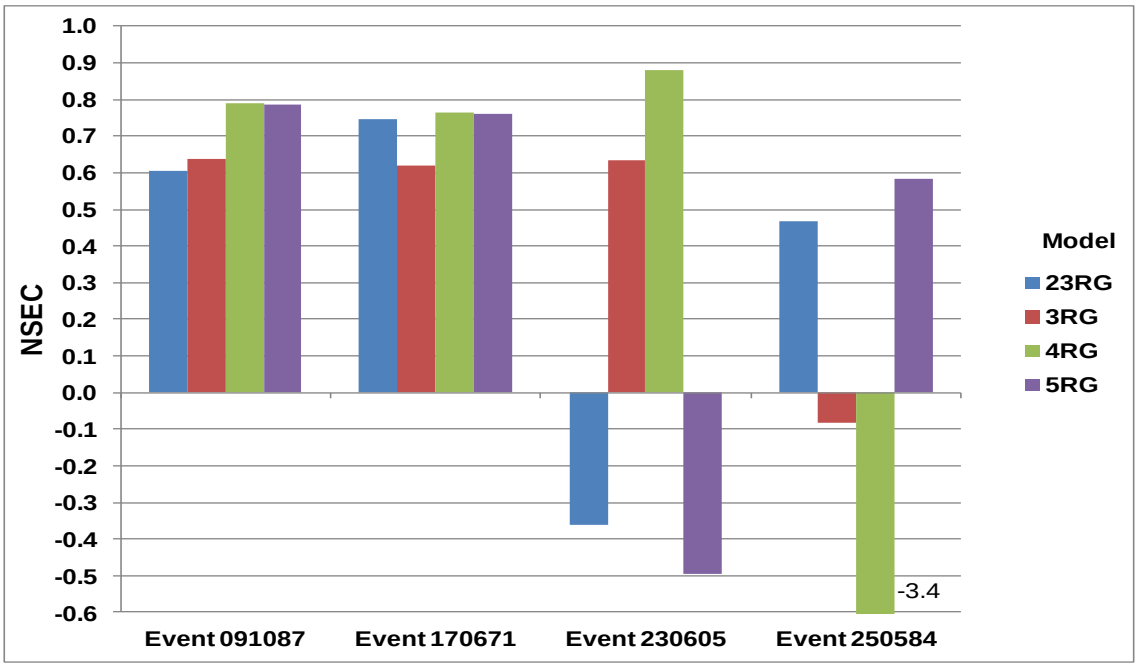


Figure 3.38. Crossness 4 models: NS scores for CSO 36786951

Figure 3.38 details the NS scores for CSO node 36786951 for the full model and 3 sub-models over four of the 5 test events for this trial. The results

for the sub-models are inconclusive. Now that the methods described in chapters 4 and 5 are available, an opportunity exists for a project to work on feature selection for the inputs, combined with multiple trials varying the architecture of the ANN to establish a reliable set of sub-models with satisfactory or good NS scores for large catchments such as Crossness.

3.8.3 Further results for real rainfall experiment/stage

It would be possible to document ANN model performance results for all three catchments using all metrics defined in section 3.6.2.3, but these have been documented in the UKWIR RTM project report (UKWIR, 2012). Readers are referred there for further presentation and analysis of performance results.

3.9 Analysis of ANN weight matrices – neural pathway strengths

This section describes early work on the analysis of ANN weight matrices, whilst investigating the structure in the parameters learnt during training. In a 1HL (1-hidden-layer) feedforward network it is possible to produce a matrix of combined neural pathway strengths from each input to each output, W_{io} by multiplying the two layer-weight matrices together:

$$W_{io} = W_1 W_2 \qquad (3.7)$$

where: W_1 is the weight matrix for the hidden layer; W_2 is the weight matrix for the output layer; and W_{io}^{ij} is the element of W_{io} describing the influence that input i has on output j, via all possible synaptic pathways through the neural network. W_1 has dimensions of $I \times H$ where I = number of input nodes and H = number of hidden units and W_2 has dimensions of $H \times J$ where H = number of hidden units and J = number of output nodes. Thus W_{io} has dimensions of $I \times J$.

There are as many neural pathways from any given input to any given output as there are hidden units. The effect of multiplying the 2 layer-weight matrices together (equation (3.7)) automatically sums the strengths of all the pathways from each input to each output. This will be discussed formally in Chapter 4.

In this section, weight matrices for an ANN model for Dorchester case study, real rainfall experiment/stage, manhole flood depths are analysed. A 2-minute timestep is used throughout. In order to reveal the structure behind these results, a subset of 9 output units is chosen for each ANN model analysed. Based on speed of response of the target hydrographs to changes in input rainfall, 3 units are selected as being "upstream" (rapid response), 3 units are selected as being "midstream" and 3 are chosen as being "downstream". Mean results from W_{io} for each group of 3 units for each of the 10-timesteps in the lagged input moving time window are computed and are displayed in Figure 3.39. In this figure, the influences from all types of ANN input signal [rainfall intensity | cumulative rainfall | elapsed time] have been merged and a mean of their combined pathway strengths is displayed.

The figures all clearly demonstrate a time-related structure to the neural pathway strengths associated with the inputs relating to each timestep, and that these are different for each grouping of the 3-nodes from upstream, midstream and downstream regions of the catchment. The increase of influence towards the inputs further in the past, beyond the -9 timestep point, suggests that it would be worth exploring use of even longer moving input time-windows to find the point of cut-off of influence and perhaps the optimal input-time window length. This is likely to be dependent upon catchment as well as position of nodes within the catchment. Observation of the smooth shape of the envelope (dashed line) described by the combined neural pathway strengths for the 3 upstream nodes (blue bars) suggests that a peak is reached at a lag of -7 timesteps. However, the bar marked X in Figure 3.39 at -9 timesteps is taller than those for -7 and -8 lags. This suggests that the last available lag in the time window is being used to compensate for the missing lags beyond -9 timesteps.

Figure 3.40 also shows a clear difference between mean influence values, through the inclusion of New Antecedent Precipitation Index (NAPI) as an additional bank of 10 inputs over the moving time-window.

Figure 3.41 to Figure 3.44 illustrate the influences of each of the 4 types of input: cumulative rainfall, rainfall intensity, NAPI and elapsed time for a trained network where NAPI is included as an input.

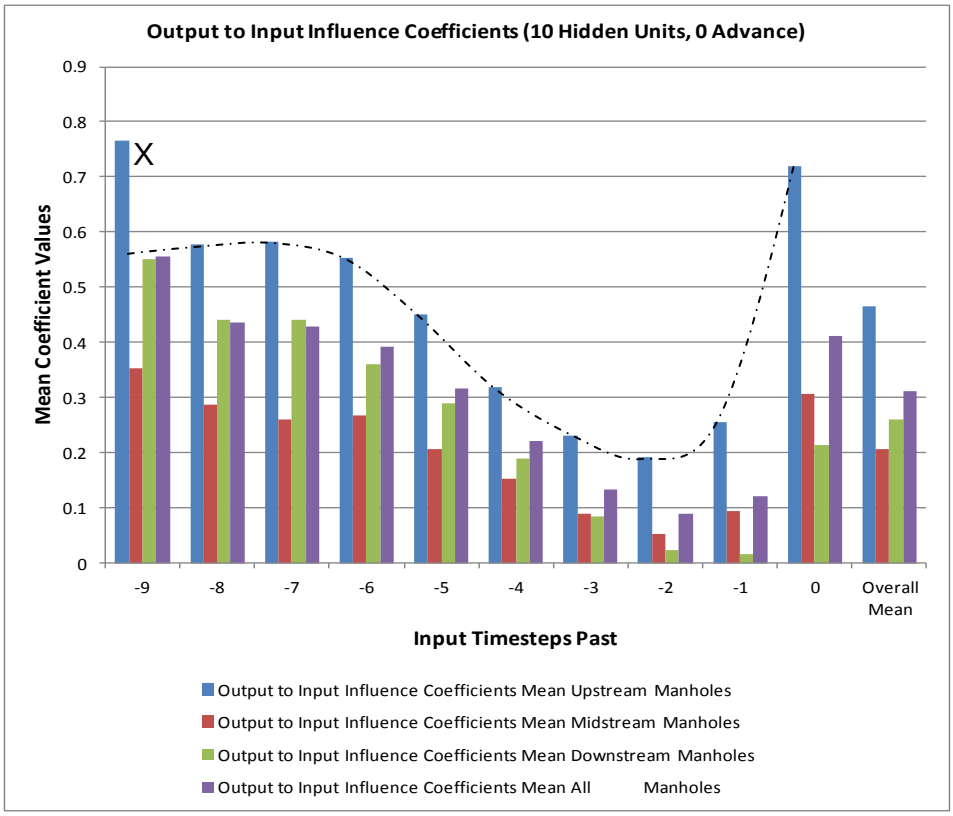


Figure 3.39. Combined neural pathway strengths for Dorchester manhole flood depth (no NAPI)

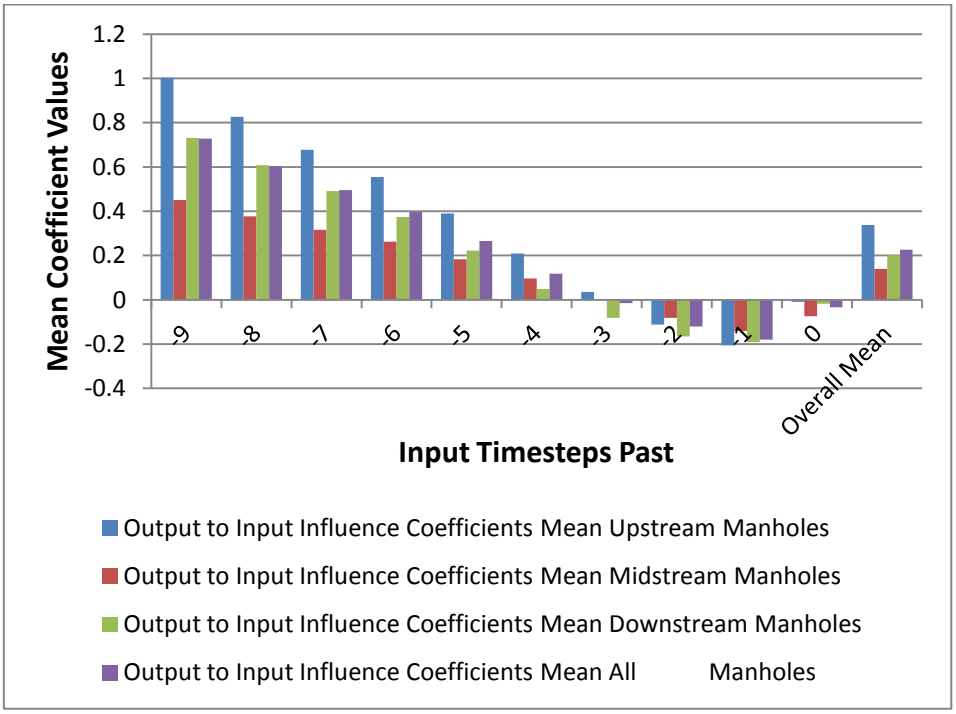


Figure 3.40. Combined neural pathway strengths for Dorchester manhole flood depth (including NAPI)

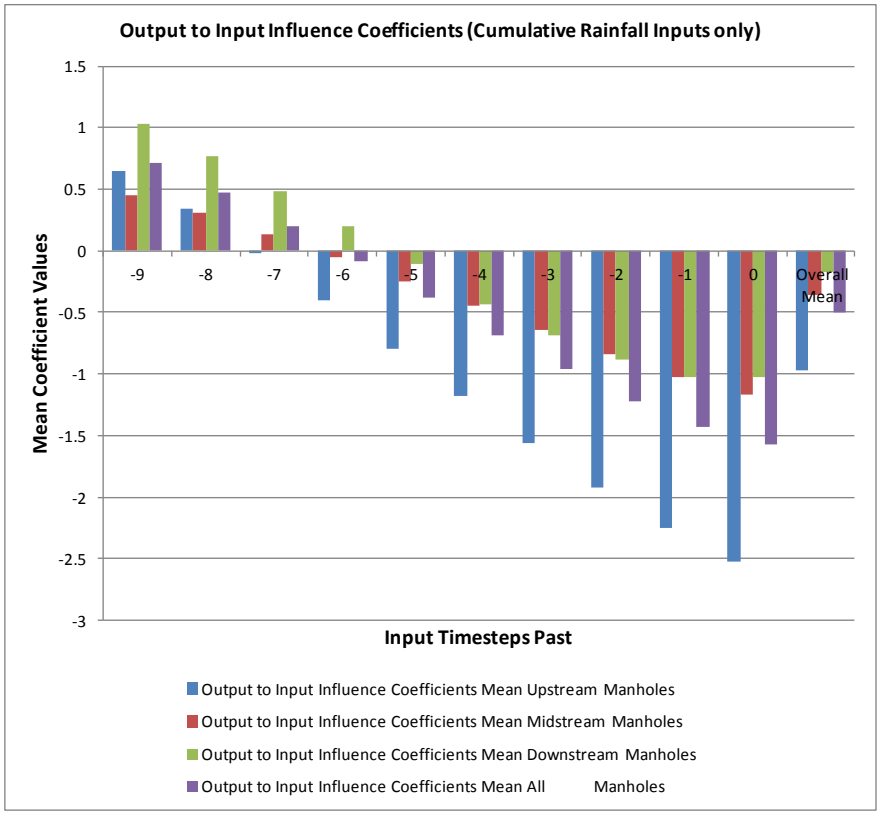


Figure 3.41. Combined neural pathway strengths for Dorchester manhole flood depth for cumulative rainfall signal inputs

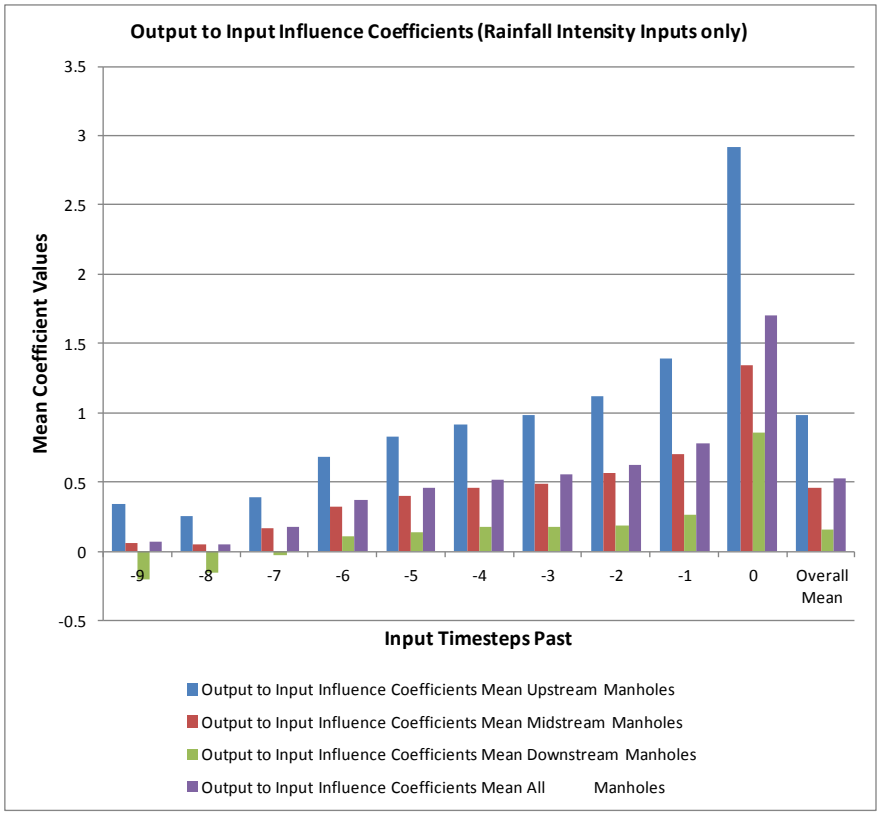


Figure 3.42. Combined neural pathway strengths for Dorchester manhole flood depth for rainfall intensity signal inputs

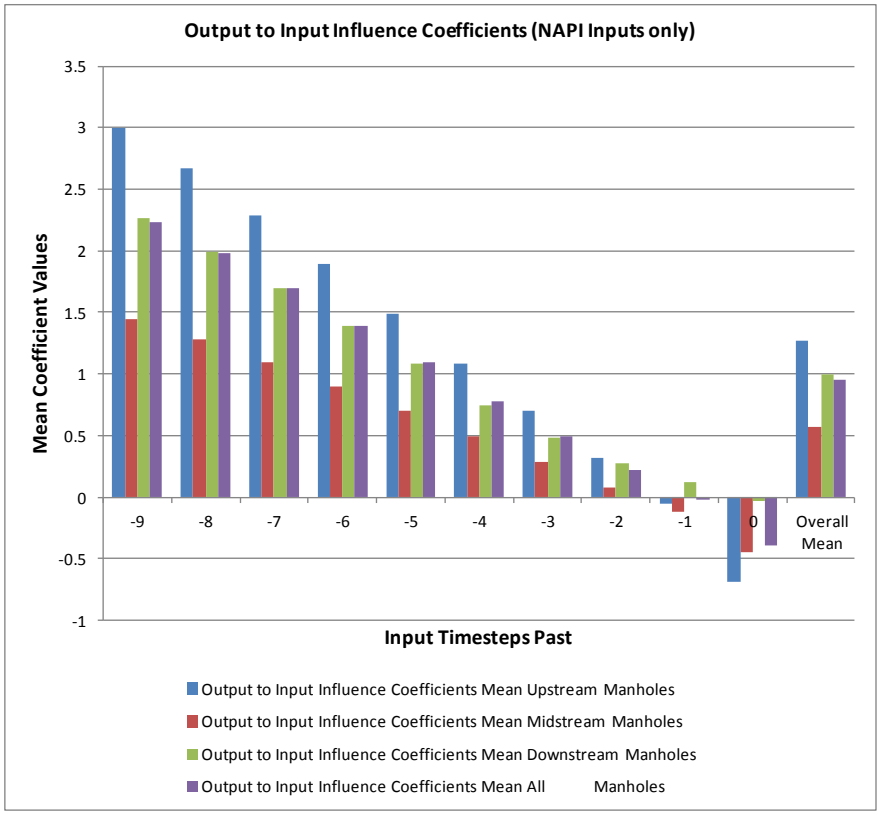


Figure 3.43. Combined neural pathway strengths for Dorchester manhole flood depth for NAPI signal inputs

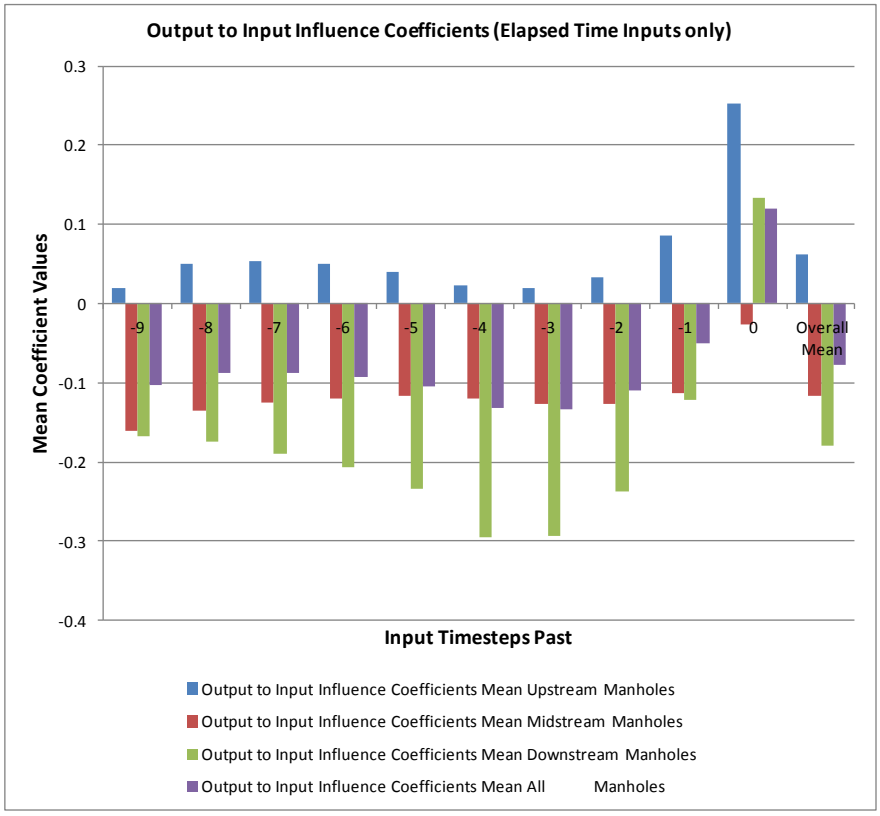


Figure 3.44. Combined neural pathway strengths for Dorchester manhole flood depth for elapsed time signal inputs

In Figure 3.41, cumulative rainfall clearly demonstrates a changeover between inhibitory and excitatory pathway strengths at different timesteps lag, depending on the position of the node grouping in the drainage network (upstream, midstream or downstream). The changeover from excitation to inhibition occurs on successive timesteps for the 3 node groupings, with upstream changing over earliest (lag = -8 to -7) and downstream is the latest (lag = -6 to -5).

Figure 3.42, rainfall intensity, reveals an opposite effect with most recent past timesteps having the largest positive influence and mean combined pathway strengths for upstream nodes being 3 times the magnitude of downstream ones. This is to be expected, since contributing areas draining to the downstream nodes are much more extensive and would have a proportionally greater effect than the local area responding to the influence of rainfall in the present and the immediate past as compared with upstream nodes.

Figure 3.43, NAPI shows that this signal (when used, at least in this case), has the largest overall magnitude of influence, showing the value of including NAPI as an ANN input. However as there is a close similarity between the NAPI and cumulative rainfall coefficient profiles, these stronger coefficients would need to be evaluated against the likely reduction in the pathway strength coefficient of the cumulative rainfall signal. NAPI and cumulative rainfall tend to have a very similar signal shape during rainfall events as can be seen in Figure 3.16, so the similarity in the ANN's treatment of these signals is not surprising.

Figure 3.44 shows the minimal influence of the elapsed time since start of each event, when used as an input. Although this is included in all the catchment case study ANN models, subsequent runs conducted without using it demonstrate that its effect is negligible on metric outcomes such as Nash-Sutcliffe. It is therefore recommended that it should be omitted from future studies.

In summary, this set of charts demonstrates that analysis of ANN weights by grouping them together as neural pathway strengths is a powerful tool to reveal structure in ANN models. It is clear that the ANN analysed here has been able to model both the spatial and temporal pattern within the drainage network and resultant set of hydrographs from different zones. The work described above sets the scene for the techniques developed in Chapters 4 and 5.

3.10 Sensitivity Analysis: Determination of the predictive limits for ANN urban flood models based on actual rainfall

3.10.1 Introduction

The material in this section expands on that published in the ICFR 2013 Conference as Duncan et al. (2013b). Many scientific papers have been produced on the use of ANNs for flood modelling and prediction. The majority of these use ANNs with a single output unit and arrange prediction for 1-timestep ahead. However, this section presents an experiment to determine the limit of predictability in terms of timesteps ahead for a multi-nodal ANN urban flood model using a moving lagged-input time window, when based on using actual (as opposed to predictions of) rainfall.

The hypothesis is that model performance will degrade rapidly for each sewer node, when trying to predict beyond its time of concentration (ToC). Furthermore prediction accuracy will be optimal when predicting at an advance equivalent to the ToC. This of course varies from node to node and typically is shorter in the upstream areas of an urban drainage network than in the downstream areas close to the wastewater treatment works (WWTW).

3.10.2 Methodology

Only the variations in methodology from that described in section 3.6 are described below; otherwise the same methodology as section 3.6 is followed:

For this case study, 10 CSOs and 6 manholes within the Portsmouth urban drainage network are used. The number of output neurons is given by the above number of key nodes (16) to be modelled in this network. The quantity to be predicted in each case is water level (also referred to as “flood depth”).

The number of neurons in the hidden layer and number of input units are varied to establish an optimum at the start of the experiment.

3.10.2.1 Input data preparation

A moving time-window lagged approach (Bowden et al., 2005; Campolo, 2003; Fernando et al., 2005; Luk et al., 2000) is implemented. A number of time-series signals (e.g., rainfall intensity, cumulative rainfall during event, etc.) are provided as inputs to the ANN. In this case study there are three input time-series: rainfall intensity (mm/hour), cumulative rainfall (mm) and the New Antecedent Precipitation Index (NAPI) value (metres) (Kellagher, 2012b) – a derived measure of soil moisture. The number of input units is given by: *number of input time-series signals (3) x number of lagged timesteps in the moving input time window*. All lags within the window are used, due to the different dependencies that may arise across all 16 model outputs because of the range of ToC's for the corresponding sewerage nodes.

The trial described is based on sixteen design rainfall events of durations from 0.5, 1, 2 or 4 hours and return periods of 1, 5, 20 or 50 years. Of these, 4 are used as test events and the remaining 12 are used as the training events. All use Laplace-distributed design rainfall intensity profiles. Table 3.4 details the design events used. This is a standard profile for design rainfall events and is particularly appropriate for simulation of summer convective storms (Faulkner, 1999; Kjeldsen, 2007), which tend to be important from the perspective of flash flooding.

Figure 3.45 is for a 1-hour duration design rainfall event of a 20-year return period for the Portsmouth catchment. This is shown highlighted as Event 14 in Table 3.4. It shows all 3 input signals (as hyetographs downwards from top) as well as target signals (as water level hydrographs) for the selection of 10 CSO's and 6 manholes used in this study. The wave-shapes of the hydrographs can be observed to be similar, yet exhibit different response times and peak profiles.

In order to evaluate the ToC's for these nodes, cross-correlations are computed between each rainfall intensity hyetograph and the corresponding 16

hydrographs for a range of delays of the rainfall signal 0 to 3600 seconds. The delays corresponding to the peak of cross-correlation are taken in the case of each node as an approximation to ToC for the event. Figure 3.46 illustrates this for the above example test event.

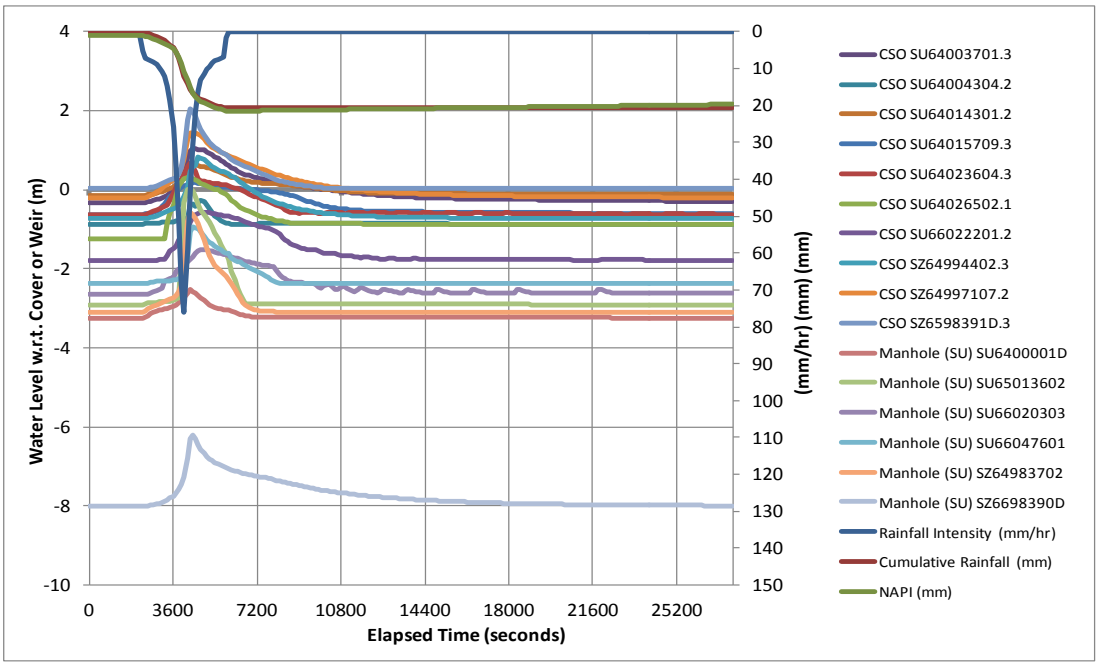


Figure 3.45. Design rainfall test event (RP=20 years; Duration=1 hour) for Portsmouth catchment

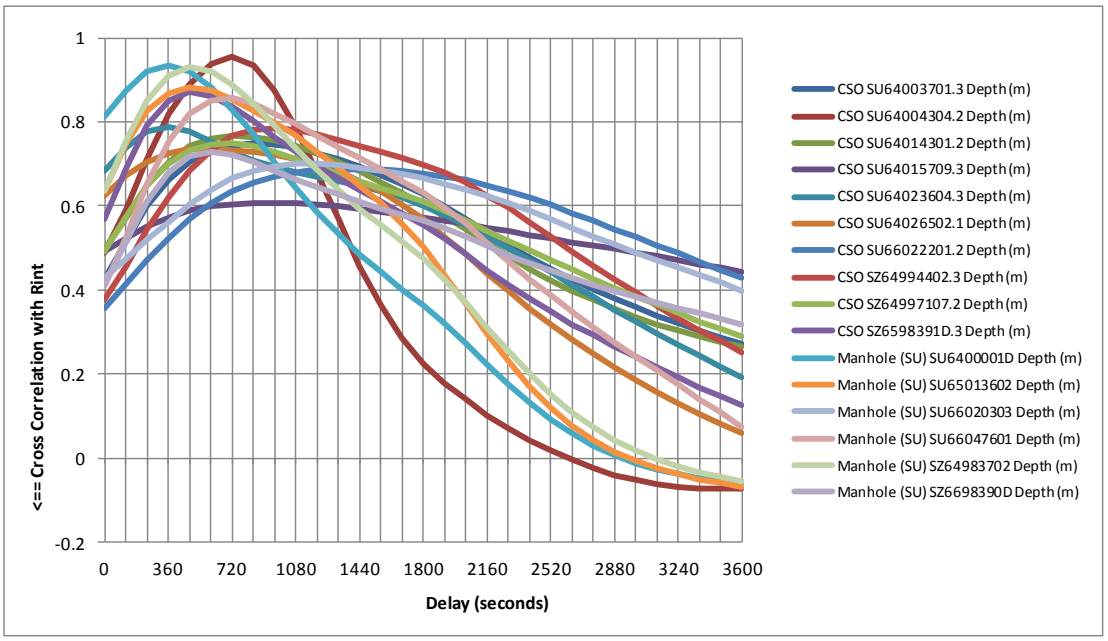


Figure 3.46. Cross-correlation functions for a set of sewer nodes over a range of delays 0-1 hour for design rainfall test event (RP=20 years; Duration=1 hour) for Portsmouth catchment

Cross-correlations are computed for all events and the spreads (over the set of 16 events) of the delay values of the peaks of these are shown in Figure 3.47 for each sewer node to be included in the ANN model. These are ranked in

order of increasing maximum delay value, which can be taken as an approximation to the true time of concentration (Butler and Davies, 2004) for each node. However, in this study we use the actual delays for each event and each node. One of the advantages of using design rainfall for this type of experiment is that the hydrographs are single-peaked. This avoids the cross-correlation method of finding ToC from being confounded by multiple rainfall and corresponding hydrograph peaks.

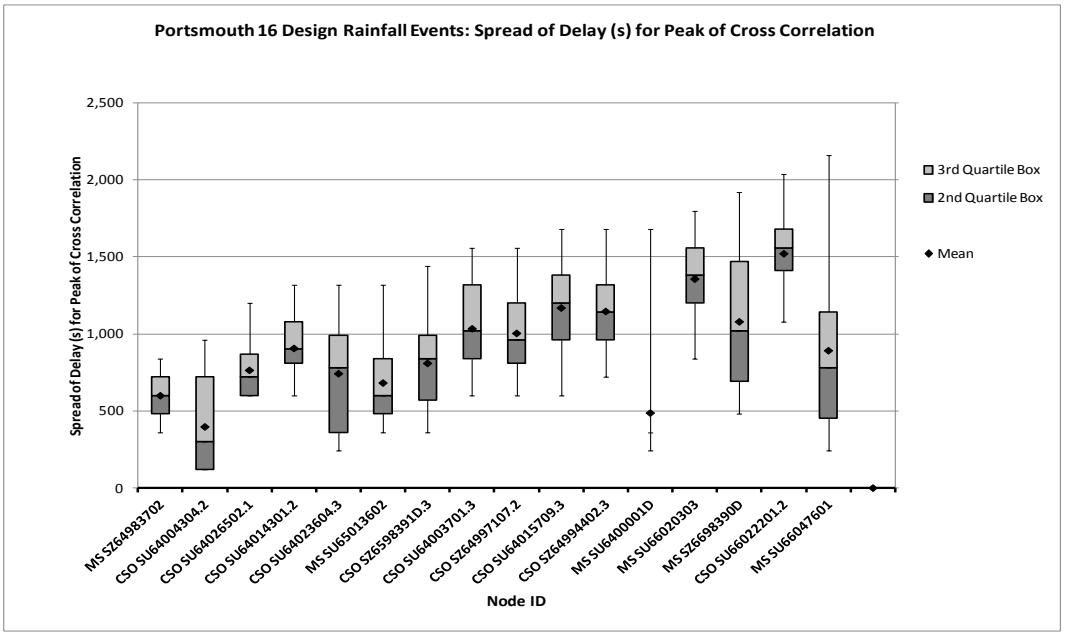


Figure 3.47. Spreads of cross-correlation peak delays (seconds) for a set of sewer nodes over 16 design rainfall events for Portsmouth catchment

For the 16 nodes from the Portsmouth catchment used in this case study, values of cross-correlation peak delay are between 6 and 35 minutes, with a median of 14.0 minutes. These are taken as indicative of the range of ToC's for these nodes.

3.10.2.2 Metrics for evaluation of ANN performance

Results using two metrics are presented. First, in order to evaluate overall performance of each ANN output unit over the first 5 hours of the hydrograph for each event, the Nash Sutcliffe Efficiency Coefficient (NSEC) is computed (Nash and Sutcliffe, 1970). The formula used is included in section 3.6.2.3. Second, in order to evaluate the combined time and amplitude error of the peak of each hydrograph a metric is developed:

$$TA_{err} = (t_t - t_m)(d_t - d_m) \qquad (3.8)$$

where: TA_{err} = time-amplitude error (metre minutes); t_t = time of peak of target (observed) hydrograph (minutes); t_m = time of peak of modelled ANN output hydrograph (minutes); d_t = water depth of peak of target (observed) hydrograph (metres); d_m = water depth of peak of modelled ANN output hydrograph (metres). This is chosen as an operationally important measure, since it is closely related to the error in predicting the impact of flooding and CSO spills. The choice of units as metre-minutes is also felt to be likely to be more operationally relevant than conversion to a percentage, for example.

The time-amplitude error TA_{err} is illustrated by the area of the shaded rectangle in Figure 3.48. In this case the ANN is shown under-predicting the peak depth (amplitude) and predicting the peak occurring 32-minutes late. This gives TA_{err} = 14.7 for a prediction advance of 30-minutes and a ToC of 14.0-minutes for this node and event; i.e. *PTA / ToC = 2.14*, which is discussed in section 3.10.3.2. It is worth noting that despite this poor performance, the NSEC for the first 5-hours of the hydrograph is 0.830, a score in the previously defined "satisfactory" class, indicating the necessity of this second evaluation metric.

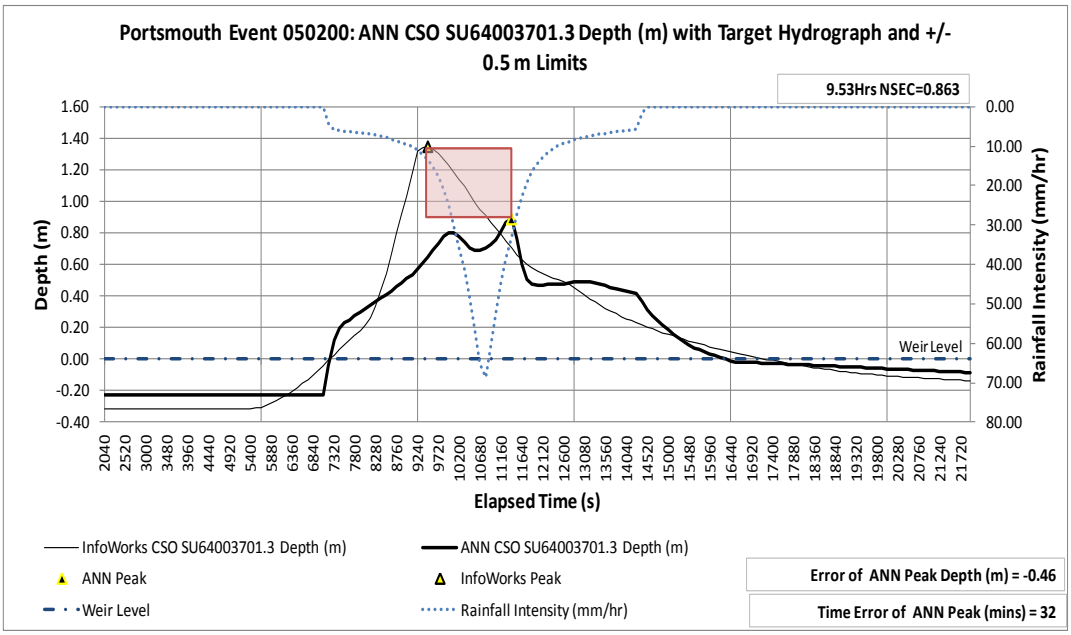


Figure 3.48. Illustration of time-amplitude error metric for 30-minute prediction advance

3.10.2.3 Optimisation of ANN architecture

A single ANN with one output unit for each of the 10 CSO's and 6 manholes is used to predict hydrograph flood/spill depths. For the architecture optimisation process, a single prediction timestep advance (*PTA=120s*) is used.

Two parameters need to be set for optimum model performance, whilst maintaining a parsimonious architecture:

- Number of timesteps lag in the moving input time window ('N_{IN}')
- Number of neuron units in the hidden layer ('N_{HU}')

A range of ANN's using combinations of values: N_{IN}*=[1,3,6,9,12,15,18,21,24,30]* and N_{HU}*=[3,6,10,15,20,30,40,60]* are trained using the same 12 rainfall events (see Table 3.4). SCG optimisation algorithm in offline batch mode as before is used. During training, the performance metric used is mean-squared error (MSE) with a regularisation term to penalise high values of sum-of-squares of neuron weights. This helps to reduce problems with overfitting (Bishop, 1995; Han et al., 2007). Early stopping is also used for the same reason. Prior to training, the ANN weights and biases are initialised to different random values, to help demonstrate robustness in the method. Following training of each ANN, NSEC scores are computed for each node and each of the 4 test events. The optimum ANN architecture is then established by looking at the spreads of NSEC values for all node outputs and choosing the combination with lowest N_{IN} and N_{HU} without significant degradation of performance. The TA_{err} metric is not used at this stage because performance at *PTA=120s* (1-timestep advance) is sufficiently good that it is unable to discriminate between good and poor ANN architectures.

3.10.2.4 Prediction timing trial

Using the optimum ANN architecture, a timing trial is then performed evaluating NSEC and TA_{err} performance for each value of prediction timestep advance (*PTA*) from 0 timesteps to 30 timesteps (1 hour). For each value of *PTA* a new ANN is trained as above and then the metric performance assessed using each of the 4 test rainfall events. Results are analysed by re-scaling the x-axis (*PTA*) as a proportion of the peak cross-correlation delay (approximate time-of-concentration) for each node and for each event. This is so that even though the predicted nodes sewer nodes have different times of concentration, the NSEC scores can be viewed as a function of prediction advances scaled in units of “time of concentration”.

3.10.3 Results & Discussion

3.10.3.1 Optimisation of ANN architecture

Figure 3.49 and Figure 3.50 show ranges of NSEC scores for a prediction advance of 1-timestep (120s) for all nodes for the shown combinations of N_{IN} and N_{HU} used in the ANN architecture. *Note: minima have been truncated at zero for purposes of the charts*. From this, values around N_{HU}=20 for hidden units performed best, over a wide range of values of N_{IN} (moving time window timesteps), suggesting that this would be a robust value to use.

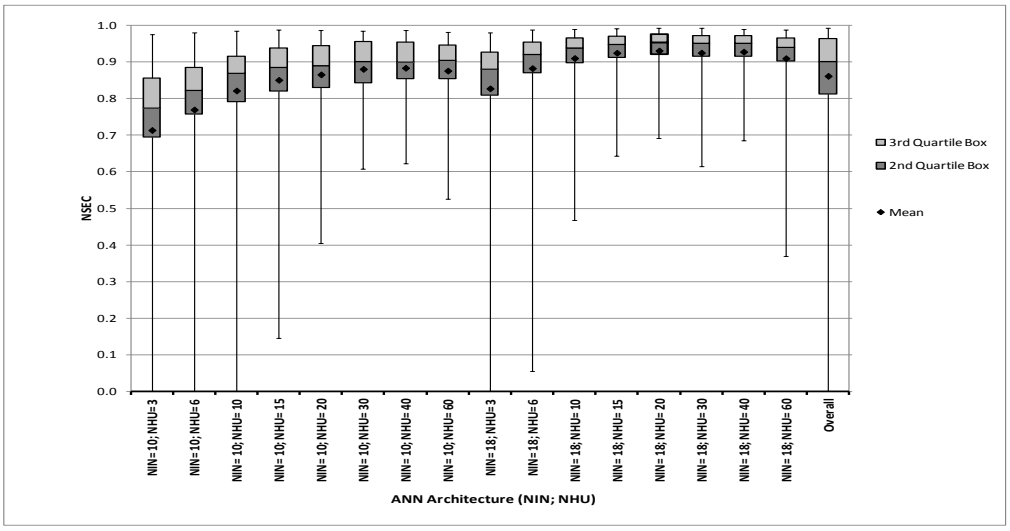


Figure 3.49. Portsmouth 4 test design rainfall events: Spread of NS scores for N_{IN}=10 and 18 and various N_{HU} values of ANN architecture

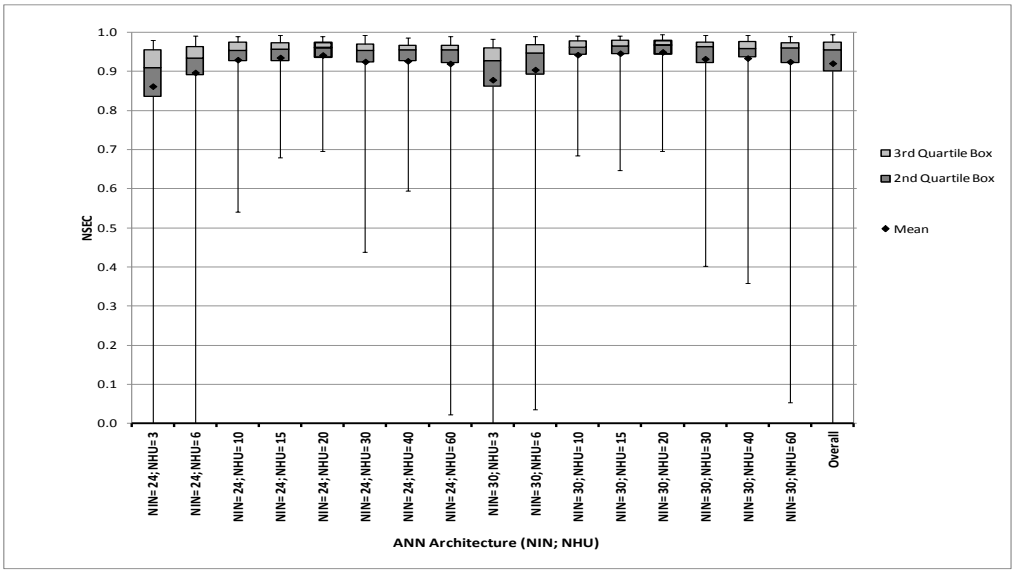


Figure 3.50. Portsmouth 4 test design rainfall events: Spread of NS scores for N_{IN}=24 and 30 and various N_{HU} values of ANN architecture

Although the ANN's using N_{IN}=30 performed best, those with N_{IN}=18 did not perform significantly worse using a 95% significance level (p=0.08). Values

of NSEC for $N_{IN}=18$ and $N_{HU}=20$ are above *0.69* in all cases, with a median value of 0.95 across all nodes and all 4 test events. Using the same NS score classification as in the earlier stages of this case study; this means all nodes fall into at least the "satisfactory" class, with the large majority of nodes in the "good" class.

3.10.3.2 Prediction timing trial

Using the optimised ANN architecture of $N_{IN}=18$ and $N_{HU}=20$, the following results are produced for the timing trial described in methodology section 3.10.2.4.

Figure 3.51 analyses the NSEC scores for each node as PTA is increased from zero to 60-minutes. It is worth remarking again that each value of prediction advance is based on a different ANN, trained by advancing the target signals by that time-interval. In the chart, the x-axis has been re-scaled to normalise to the ToC for each node, such that an x-value of 1.0 is for *PTA = ToC* for that node. The ToC is measured by the peak of the cross-correlation (*Xcorr*) delay for each event and each node. Values above 1.0 are for prediction advances greater than time-of-concentration for the node and *vice versa*. The figure shows the results for the 5-year return-period (RP), 2-hour duration design rainfall event.

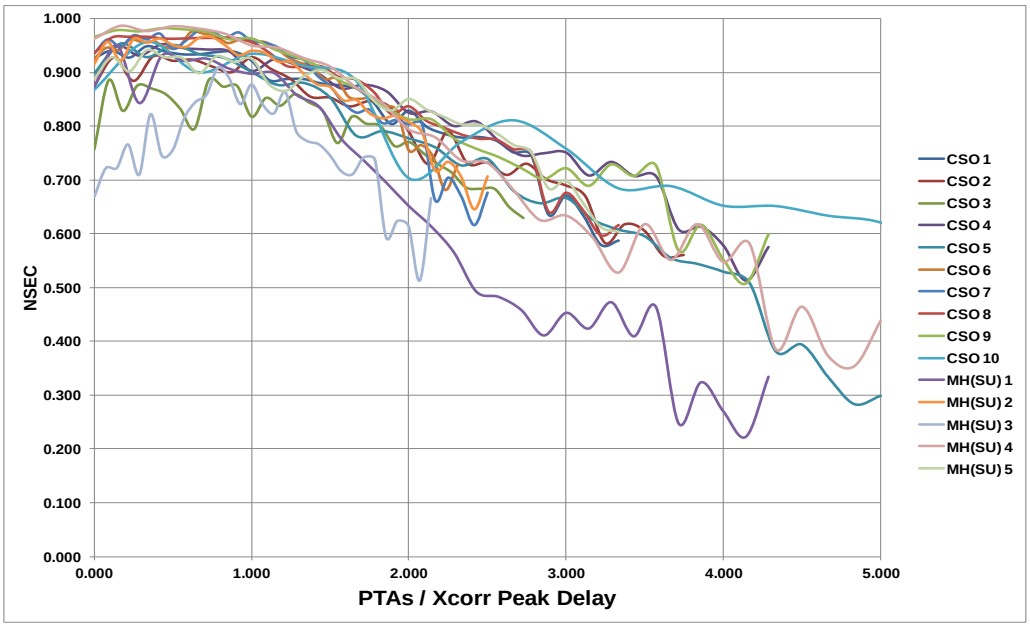


Figure 3.51. NSEC scores versus ratio of Prediction Timestep Advance to ToC for 5-yr RP, 2-hr event for 16-nodes from Portsmouth catchment

This clearly demonstrates satisfactory or good performance for PTA less than or equal to 1.0 x ToC and degrading performance for prediction advances above this value. In some cases, as hypothesised, NSEC performance actually improves towards *PTA=ToC* then degrades again above this level. This is to be expected, since when *PTA=ToC*, the peaks of rainfall and hydrograph are perfectly synchronised, making the simplest possible relationship between ANN inputs and outputs. In other words, the ANN is most capable of predicting flooding at manholes at a prediction advance, which is equal to the arrival time of the water at the manhole.

In order to demonstrate that the methodology for the normalisation of the x-axis to ToC does indeed reveal the underlying structure in the NS results, the same data as in Figure 3.51 are presented in Figure 3.52. However, here the x-axis is simply scaled in seconds of prediction advance, regardless of output unit.

As can be seen, the structure in the data is still present, but not so clearly revealed as when the prediction advance is scaled as a proportion of ToC for each node. Results for the other 3 test rainfall events then follow in Figure 3.53 (1-year RP, 1-hour duration), Figure 3.54 (20-year RP, 1-hour duration) and Figure 3.55 (50-year RP, 2-hour duration).

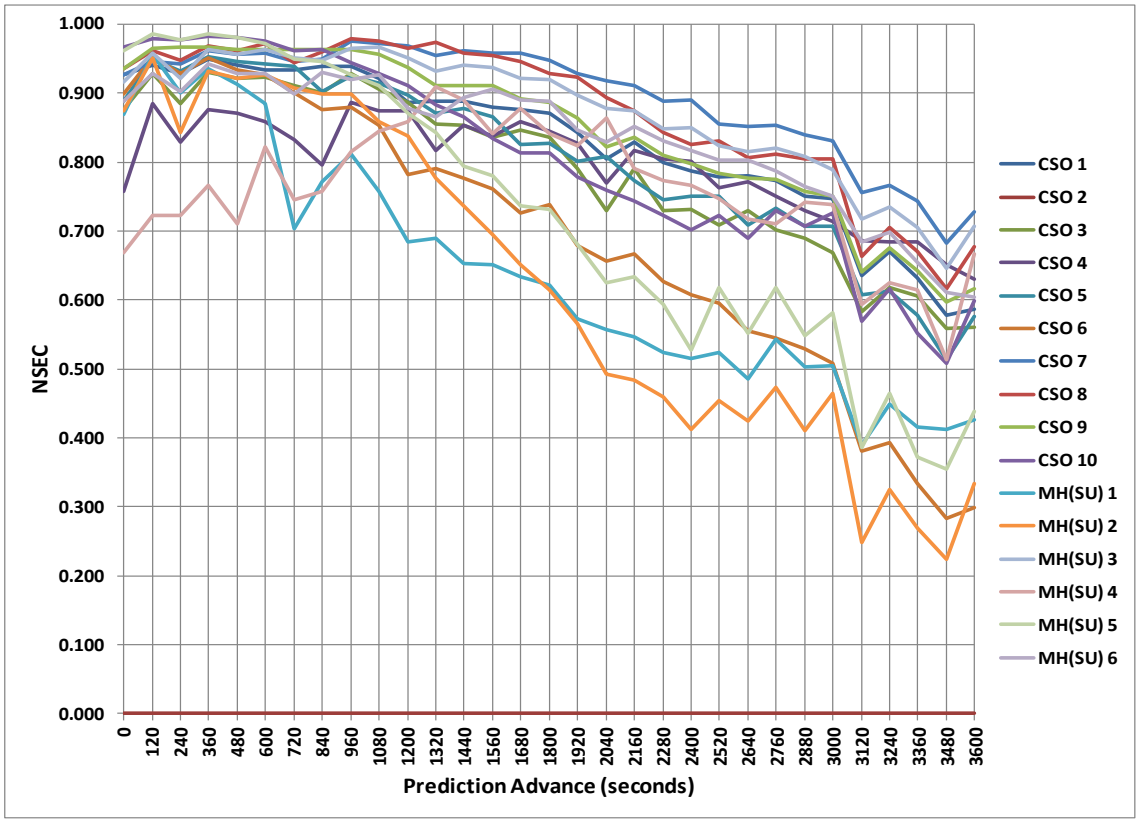


Figure 3.52. NSEC scores versus Prediction Advance (seconds) for 5-yr RP, 2-hr event for 16-nodes from Portsmouth catchment

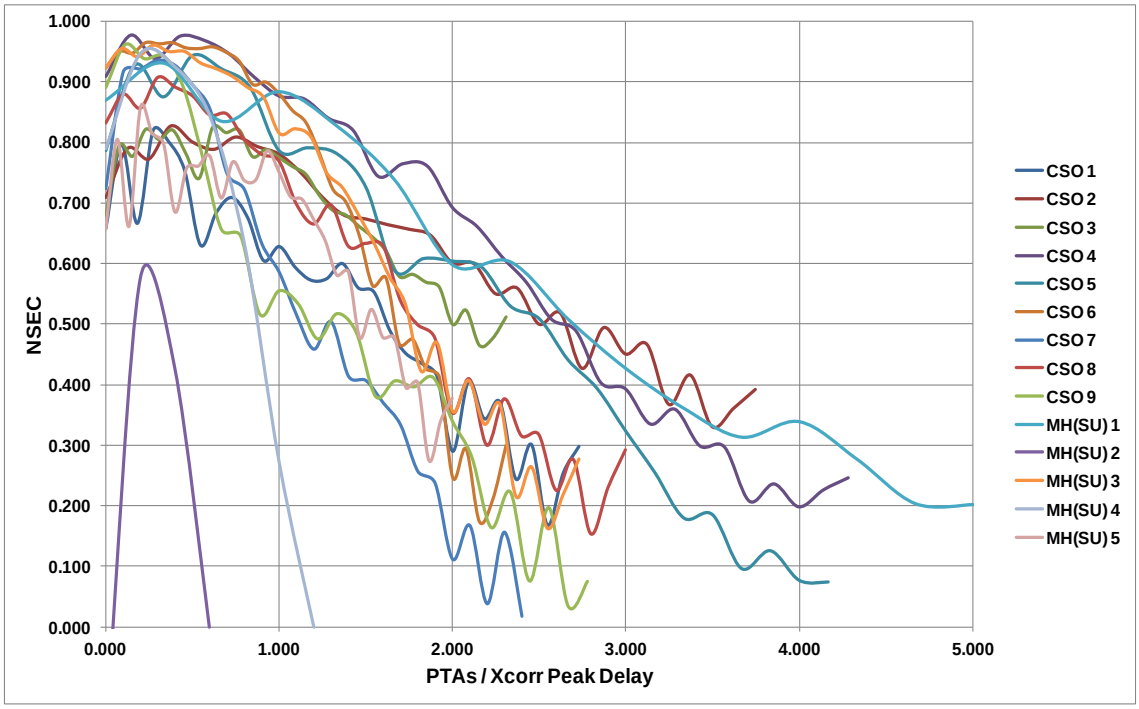


Figure 3.53. NSEC scores versus ratio of Prediction Timestep Advance to ToC for 1-yr RP, 1-hr event for 16-nodes from Portsmouth catchment

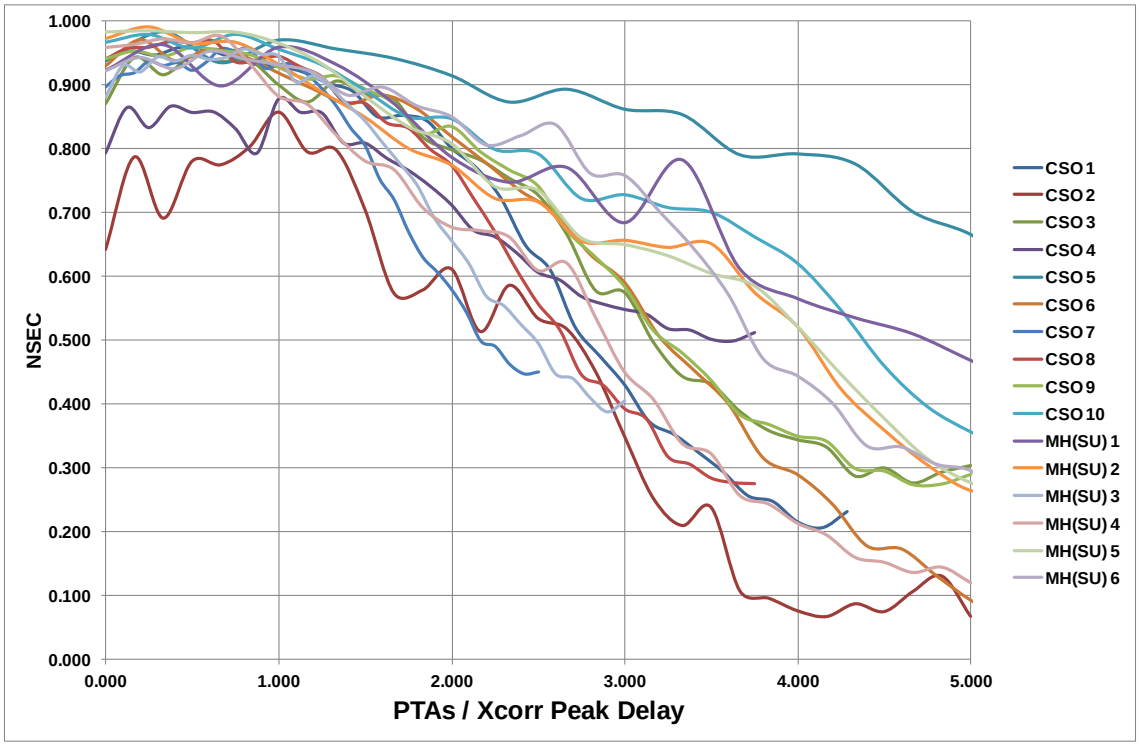


Figure 3.54. NSEC scores versus ratio of Prediction Timestep Advance to ToC for 20-yr RP, 1-hr event for 16-nodes from Portsmouth catchment

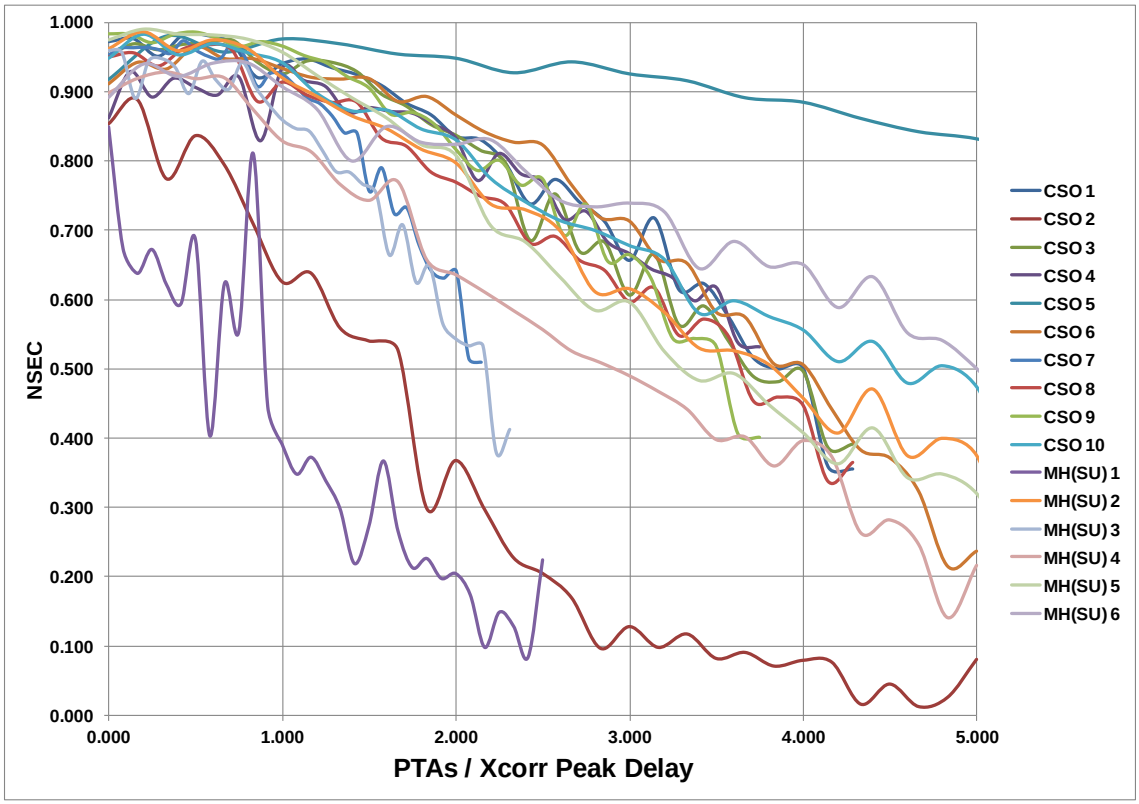


Figure 3.55. NSEC scores versus ratio of Prediction Timestep Advance to ToC for 50-yr RP, 2-hr event for 16-nodes from Portsmouth catchment

From Figure 3.53 and Figure 3.55 for the least and most intense rainfall events, it can be seen that NS performance starts to deteriorate for some nodes even before the *1.0 x ToC* prediction advance point is reached. Conversely, for the moderate intensity events, shown in Figure 3.51 and Figure 3.54, NS performance either remains as a reasonably constant high level up to the *1.0 x ToC* prediction advance point, or even improves for some nodes up to that point. In all cases, NS scores fall off rapidly beyond *1.0 x ToC* prediction advance, as hypothesised and show that the time of concentration represents a limit on the predictive capability of the system.

Time-amplitude error metric (TA_{err}) results

Figure 3.56 presents a similarly-formatted chart for the second metric: time-amplitude error (TA_{err}) for the 5-year RP, 2-hour duration rainfall event. This shows an even clearer degradation of performance for PTA above *1.0 x ToC*. Figure 3.57 is for the 1-year return period, 1-hour duration event and again illustrates good performance for *PTA<=ToC*, but demonstrates the tendency for ANNs to over-predict less severe events as its performance breaks down for the longer prediction advances.

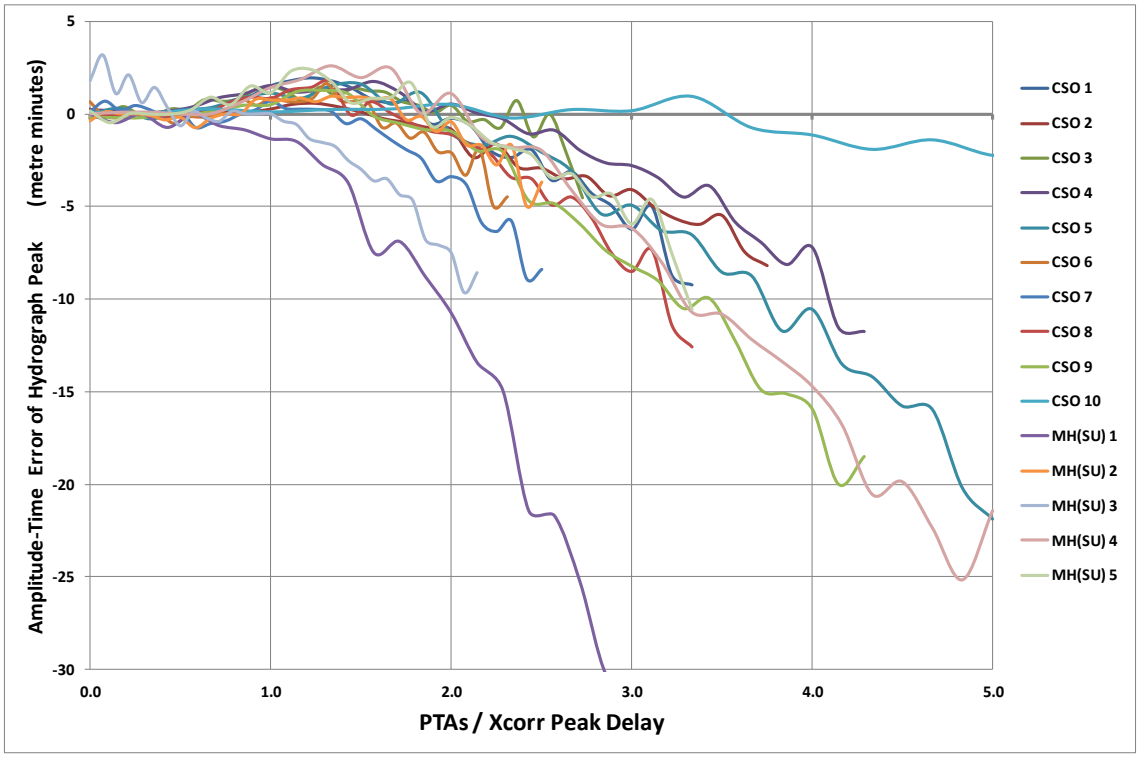


Figure 3.56. TAerr scores versus ratio of Prediction Timestep Advance to ToC for 5-yr RP, 2-hr duration event for 16-nodes from Portsmouth catchment

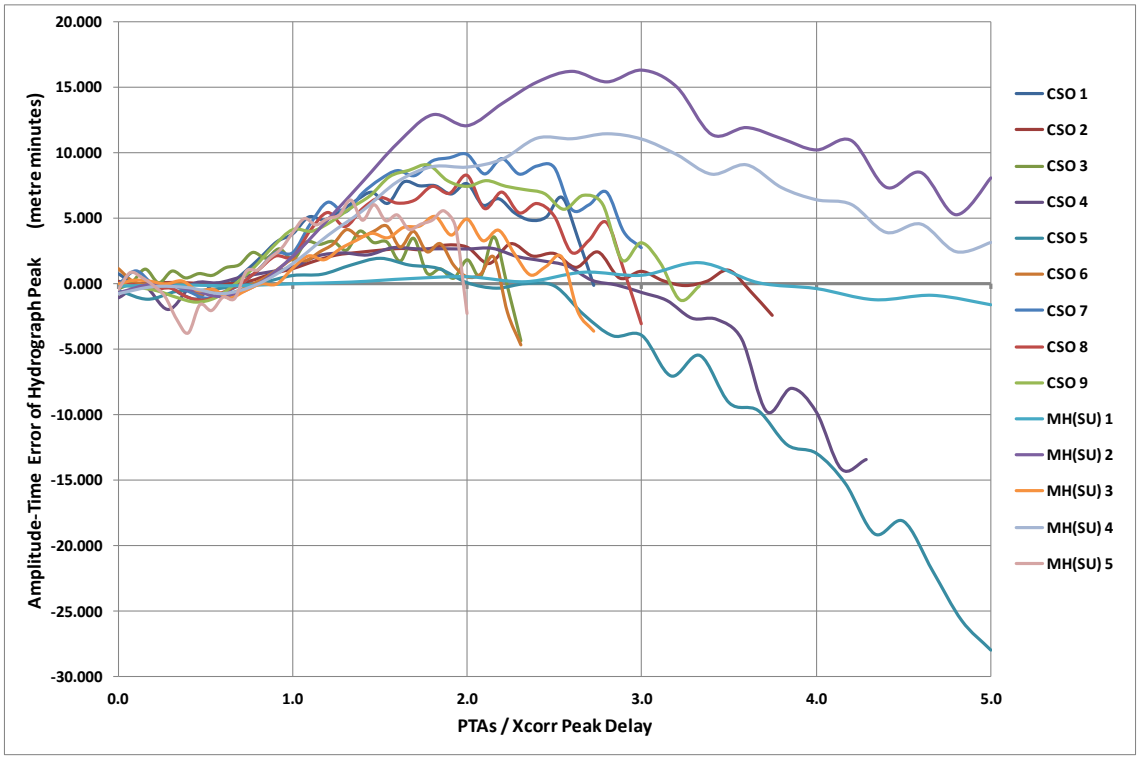


Figure 3.57. TAerr scores versus ratio of Prediction Timestep Advance to ToC for 1-yr RP, 1-hr duration event for 16-nodes from Portsmouth catchment

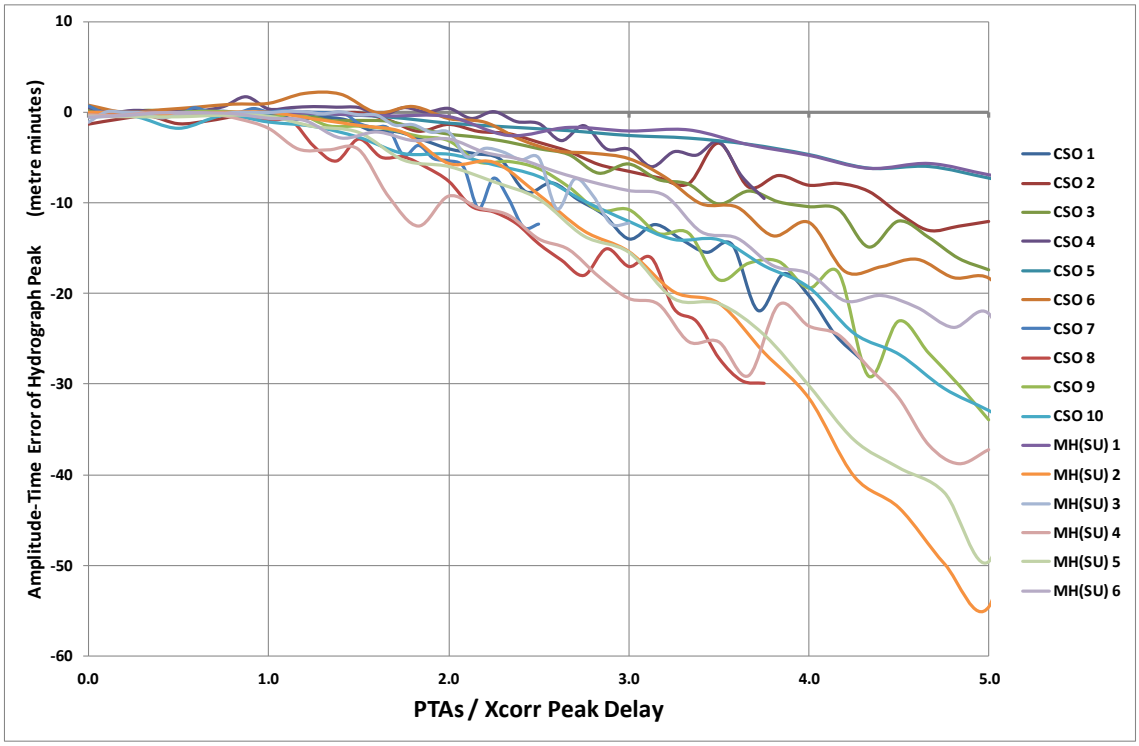


Figure 3.58. TAerr scores versus ratio of Prediction Timestep Advance to ToC for 20-yr RP, 1-hr duration event for 16-nodes from Portsmouth catchment

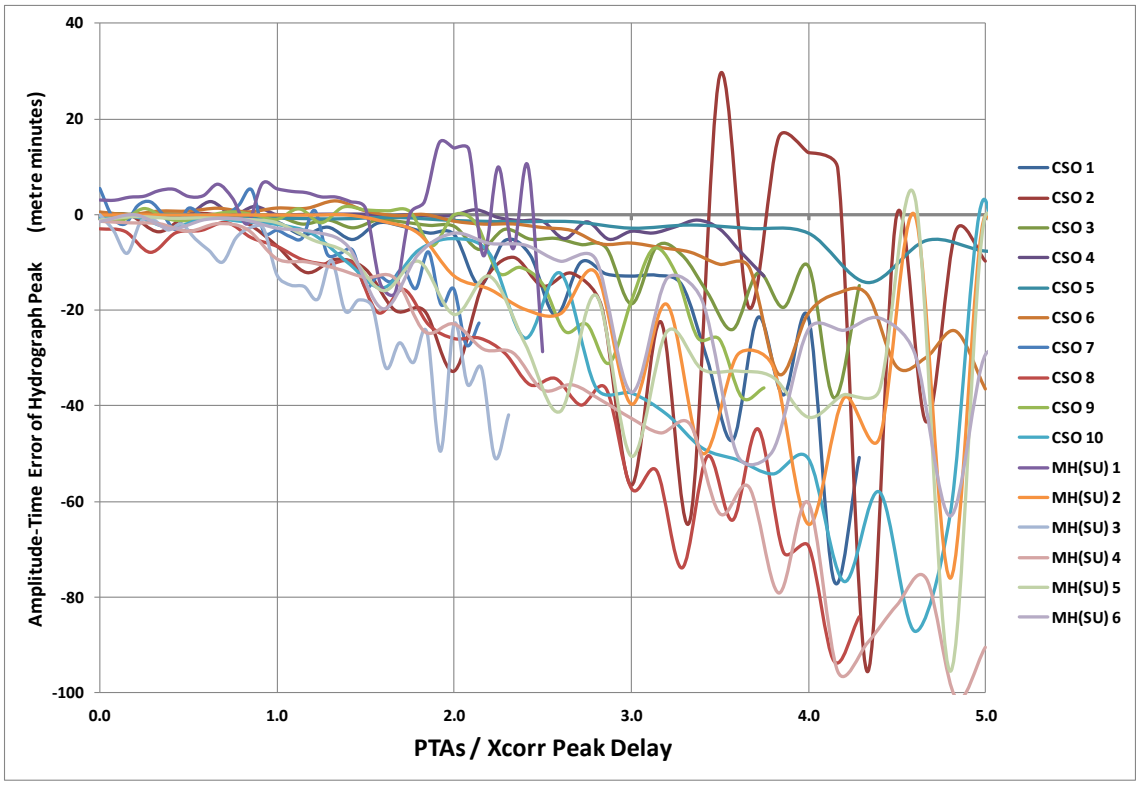


Figure 3.59. TAerr scores versus ratio of Prediction Timestep Advance to ToC for 50-yr RP, 2-hr duration event for 16-nodes from Portsmouth catchment

Figure 3.58, for the 20-year RP, 1-hour duration event and Figure 3.59, for the 50-year RP, 2-hour duration event also present a similar pattern to the NS results, with the last figure demonstrating a tendency for the peak flooding of the most intense rainfall events to be under-predicted, beyond advances of ToC.

3.11 Discussion: Use of ANN models for urban flooding

A range of key contributions have been made during the process of this case study that establish advances in urban flood prediction and modelling. This section discusses these advances in understanding and provides some conclusions.

3.11.1 Multi-output ANN model performance

This case study explores in depth possibilities for using multi-output ANNs to model several nodes in a sewer network simultaneously. These models exploit the similarities between hydrograph shapes in order to construct individual predictions for flood depths, volumes or flow rates at each modelled node. Each ANN output unit uses the same set of neurons on the hidden layer at different relative weights, to provide predictions at an individual sewer node. From research in the literature, this appears to have been done very little before, if at all, yet is proposed here as a logical approach to the requirement of rapid real-time flood prediction at multiple locations in an urban drainage network.

The first ANN runs trained on the initial data were found to produce poor performance. An investigation conducted as to the reason for this identifies two primary causes:

Initially, a single ANN model is used to try and replicate all the hydraulic signals for all categories and for all locations. The training data contains target data signals of depth and volume measurements for manholes and CSOs respectively and also uses cumulative values of volume as well as rates of flow (volume / unit time). Output levels are defined in the measurement units appropriate for each output unit. Since linear transfer functions are used on the output network layer it is possible for ANN outputs to be calibrated in these units

of measurement by setting the output layer weights accordingly. However, this means that very different ranges of values for depth (even between CSOs and manholes) and between depth and volume nodes results in too great a dynamic-range between the smallest and the largest signals. This results in unnecessarily high levels of error due to excessive complexity in such a combined model.

The recommended solution guidelines are:

- To avoid using certain inputs (cumulative volume in particular is problematic); instead, volumes per timestep are used.
- To avoid modelling cumulative quantities; but instead use a post-processor to sum the ANN responses at each timestep during each rainfall event;
- To use separate models for different measurement unit types (e.g. volume and depth) – although volume (m^3/timestep) and flow (m^3/s) have been modelled together successfully;
- Where possible consider modelling different node types using different ANNs (e.g. manholes and CSOs) – the levels for these are very different, with water levels in manholes being measured in metres and spill depths for CSOs being measured in fractions of a metre. Best accuracy can thus be attained by modelling manholes and CSOs separately – although this case study includes examples of successfully modelling them together with acceptable results.

Following the above guidelines has resulted in greater success with the accuracy of models' prediction corresponding with the "observed" test hydrographs.

3.11.1.1 Separate ANN per sewer node versus multi-output ANNs

A number of models are run using "multiANN" mode, in which a separate ANN with a single output unit is created for each target sewer node location. These demonstrate generally rather small improvements in performance over the single multi-output ANN models. However, although in principle a separate model could be trained on each location with two models used: one for predicting depth and the other for predicting volume, there is a significant

additional computational demand in training multiple ANNs compared to using a single model for all locations for any specific node type and measurement category. The parsimony achieved by exploitation of similarities in the hydrograph response shapes through re-using the same hidden layer for all outputs in a multi-output ANN would be lost. The training time saved by the use of multi-output ANNs would also be lost.

On the other hand, the used of single-output ANNs would make the task of input feature selection for the optimal and most parsimonious model easier. For example, the number of lagged input timesteps needed as a function of the location of the point of interest in the network (upstream or downstream) would be able to be optimised separately for each node, rather than needing to take all nodes being modelled into account together.

The problem of different ranges of measurements (flood volumes say of 10's or 1000's m^3) would also not be an issue if modelling each node with a separate ANN model.

3.11.1.2 ANN output clamping

Where target signals are provided so that values are constrained to not exceed a maximum or minimum value (for example minimum of volume is 0; maximum of manhole flood depth is 0) the same approach is also implemented for the ANN output. This is achieved again by post-processing the ANN output hydrographs to the given limits. This has the effect of improving Nash-Sutcliffe (NS) and scores for other metrics, so it is felt to be a valid approach, since the model output values that are being clamped-off are those that are known *a priori* to be non-valid.

3.11.2 ANN configuration and setup

3.11.2.1 Portsmouth and Dorchester

The Portsmouth and Dorchester ANN networks have a nearly identical target dataset, consisting of 40 input units (covering 4 different input variables [*elapsed time* | *rainfall intensity* | *cumulative rainfall* | *NAPI*] over a 10 timestep input window), 10 hidden units and ‘n’ output units, where n is the number of

sewerage nodes being predicted. Where NAPI input is not used, the number of input units is reduced to 30. As has also been previously noted, the elapsed time inputs are not required. This number of input features is manageable and results in satisfactory or good NS scores for most nodes. Although, for real rainfall events, the models' NS scores are lower, a majority of nodes still perform with "good" or "satisfactory" performance. In a live real-time EWS, there would undoubtedly be focus on hydrograph peak levels. Looking at the peak depth category classification results (M_{C1}, B_{A1}, M_{C2} *and* B_{A2}) shows that there is scope for accurate classification of peak levels despite not necessarily achieving perfect NS scores or hydrograph response shapes.

3.11.2.2 Crossness

The ambitious task of modelling the Crossness urban catchment has been undertaken. The challenge is the use of spatially variable rainfall (both intensity and cumulative at 23 raingauge locations of each) and spatially variable NAPI (at 40 locations), resulting in a very large number of input features. Again a 10-input timestep window is used, but an ANN architecture of 100 hidden units is found to perform better than the 10 used for uniform rainfall.

For this model the total number of input weights is, $N_{W1} = ((2 \times 23) + 40 + 1) \times 10 \times 100 = 87{,}000$. This has the effect of creating a vast high-dimensional weight-space in which to search for optimum solutions during the training. Almost any search strategy would be unfeasibly sparse without further dimension reduction and feature-selection being used. Despite this, the ANN models some nodes at the "satisfactory" level of NS score.

The broad consequence of this is that the Crossness ANN performed relatively poorly compared with the other two catchments. The early results presented for the 3, 4 and 5-raingauge sub-models, suggest that subdivision of models for very large catchments like Crossness could be appropriate.

3.11.3 Number and characteristics of training and test events required

The effectiveness of ANN models is dependent on a range of aspects, but one which is critical is to provide sufficient and relevant events on which to train the model. Some guidelines have emerged:

How to define what is a relevant training event

- Clearly there is a complex relationship between rainfall intensity, immediate antecedent rainfall, and response of a target location. It is important to recognise that the ANN model is basically non-linearly correlating shape of the target response with the shape of the driving input parameters. This means that a “relevant” training event is not necessarily to be considered in typical engineering, hydrological and hydraulic terms.
- From investigations of test rainfall events with outlying (poor) performance results, it emerges that these tend to lie outside of the envelope of cumulative rainfalls versus elapsed time for the collection of training events used. Figure 3.15 shows an example of such an envelope of cumulative rainfall for a set of training events for Dorchester. In this, one of the test events (201126) exceeds the upper edge of this envelope.
- In general training events need to be included that exceed (in both directions) the cumulative rainfalls and instantaneous intensities of those events that are expected to need to be modelled. This gives a challenge, since it may be required to model unprecedented events at some future time, in a live EWS. Strategies for artificially augmenting rainfall intensities of extreme events used for ANN training may be required.

How many training events are needed to achieve a good ANN model

- An examination of the results obtained does not provide any obvious conclusions. However what is clear is that spatial rainfall with multiple inputs requires a combinatorial increase in the number of events needed in line with the degree of “spatiality” of the catchment. For example frontal systems may advect from different points of the compass and a number of different models may be required to handle this. Convective events may need yet different models. This has implications for both limiting the size of the catchment modelled or the sub-section of catchment in each model and the types of events for model accuracy.
- In the experiment/stages of this case study using design rainfall, 11 events are used for training, 1 for validation during training and 4 are used for test; a ratio of approximately 3:1. Each event has between 300 and 900 samples.

- In the real rainfall stage, 44 events are used for training, 1 for validation during training and 5 are used for test; a ratio of almost 9:1. The same sample counts apply here too.

How to measure how good a trained model is

- Although the four or five test events used provide an indication of the accuracy of the model, in hindsight it might have been very useful to have assessed the accuracy of the ANN model against the training events as well. This is possible using an N-fold cross validation (NFCV) methodology (Kohavi, 1995), described in detail in chapters 4 and 5. A separate ANN model is built for each data-fold (here equivalent to rainfall event) and tested on that fold, having been trained on the other folds. In this way it is possible to use every rainfall event as a "test" event – at the cost of creating and training as many ANN models as data folds.

3.11.4 Other ANN configuration details

Optimisation strategies

The optimisation algorithm adopted throughout this case study is the Scaled Conjugate Gradients (SCG) method. This is described in literature review section 2.2.5. For very large weight spaces (e.g. ~100K weights, as is the case with Crossness) a further strategy for dimension reduction would be needed in addition to exploring alternative algorithms that populate the weight space with much larger sets of candidate solutions.

It would also be worth revisiting standard backpropagation (Hecht-Nielsen, 1989), as this is reported to be a robust technique in reasonably large weight spaces and is also relatively efficient, since it does not involve calculation of large matrices; instead it operates on local data at the level of each neuron.

Potential for use of GA / EA optimisation

The use of a genetic algorithm (GA) or evolutionary algorithm (EA) to optimise ANN weights and biases is only possible with the availability of considerable computational resources, as each objective function evaluation requires a full neural network run to be conducted for each member of the

algorithm's population. Even with this resource, there is the potential for overfitting the training process by this method or, in the case of very high-dimensional weight spaces, failing to locate the sub-region containing the optimal solution at all without requiring populations of millions of candidate solutions. This would be computationally unfeasible.

A preliminary experiment with the use of EAs to optimise the architecture for best performance shows that the architectural parameter settings used in this case study for the non-spatial models are close to their optima. However, different settings also provided similar results, indicating that the ANN models are quite flexible and insensitive to architecture variations. The results from the sensitivity analysis stage suggest using slightly longer input time windows and more hidden units may be of benefit for fine-tuning model performance.

Training error metric

Two training error metrics (MSE and MSEREG)(Mathworks, 2012) are employed for the Real Rainfall Experiment/Stage of the project. Both use mean squared error evaluated batch-wise over the entire 44 training events as a single scalar metric for each ANN output at the end of each training epoch.

The MSEREG training error metric is also used for several ANN model training runs. In addition to evaluating MSE, this implements weight decay regularisation to penalise high values of sum of square of weights. This has the effect of regularising the network to a low mean value of weights, thus reportedly reducing the probability of over-fitting. This is found to be particularly effective for depth hydrograph modelling, which tend to have smooth shapes. MSEREG is found to be less effective for volumes or flows, which typically have a much spikier time-domain profile. The regularisation parameter is normally set to 0.5 to give equal importance to the error and the weight penalty terms.

Some initial results when using Nash Sutcliffe Efficiency Coefficient (NSEC) directly as a training error metric are available, but are not reported here, due to lack of space.

It seems that the most important part of the flood hydrograph to model accurately is the peak, during which the majority of impacts of flooding are

experienced. Therefore it would be worth developing new metrics that reward accuracy in this part of the hydrograph most, then evaluating ANNs using them as performance functions during training. Perhaps it may be possible to aggregate the timing-amplitude error (TA_{err}) metric used in the sensitivity analysis with the MSE or NSEC metric during training. This may improve prediction of the highest impact events and avoid the effects of under-prediction of the highest hydrograph peaks.

3.11.5 Sensitivity analysis for multi-output ANN models of urban flooding

The results of the sensitivity analysis stage of the case study (section 3.10.3) clearly demonstrate that acceptable performance for multi-node urban flood prediction can be achieved using single ANNs. They are able to exploit the similarities between the flood response hydrographs at the various nodes as illustrated by an ANN with 16 output units operating well with as few as 20 hidden units. At the same time, they are able to accommodate a range of times of concentration (ToC's) for the modelled nodes, which in this case spans a range of delays from 6 to 35 minutes, a ratio of 1:6.

Corani and Guariso (2005) use pruning of hidden neurons to analyse the effect of each one effectively specialising in modelling different aspects of the overall ANN response. It is possible that use of this technique could further reveal the structure of these multi-output ANN models. Chapters 4 and 5 contain some visualisation techniques potentially relevant to such a study.

3.11.5.1 Limits for prediction advance when using actual rainfall as input

The timing trial within the sensitivity analysis stage also clearly shows that use of lagged-input feedforward ANNs based on actual rainfall (instantaneous intensity and cumulative rainfall during each event) as input signals is limited to prediction advances not greater than ToC for each node modelled. Beyond advances of *1.0 x ToC*, performance using both NS and the new TA_{err} metric rolls off rapidly. A physical explanation for this is that ToC is the length of time it takes for rainfall on the furthest (upstream) part of the catchment to arrive at the node. Effectively, trying to predict flooding beyond this advance means that

there is no longer any relevant information in the actual rainfall input signals, since the relevant rainfall will not have started yet.

Because ToC's for urban drainage tend to be less than the required 2-hours for operationally useful forecasts (Einfalt et al., 2004), it will generally be necessary to use predictions of rainfall (nowcasting) in order to achieve these. However, such models have an opportunity to augment prediction capability by using the ToC times described in this paper, which in the case of large catchments such as Crossness may be a significant advantage.

3.12 Further remarks and future work

Valuable lessons have been learnt in the process of developing multi-output ANN models for urban drainage systems. Indications are that ANN tools are generally good and computationally efficient for prediction of flooding in urban drainage systems to a level of accuracy which the water industry would find useful.

ANN models currently require expert academic input to build, train and run. However, the models have been shown to be sufficiently flexible to suggest that generic models in an executable form could be developed and provided for the water industry to use without recourse to academia.

Now that the feature selection techniques described in chapters 4 and 5 are available, there is an opportunity to revisit these models (particularly Crossness) in the context of the spatio-temporally varying signals they use and the search for more parsimonious and accurate models. As an alternative, a strategy for creating a range of sub-models with different numbers of inputs, hidden units and outputs could be investigated.

It may also be possible to extend the use of the ANN models to the nowcasting (short-term prediction up to 6-hours) of local rainfall based on rain radar images. This would then potentially allow these models to be cascaded with those already described to provide operationally useful predictions of flooding. Early results obtained in this regard show promise, but are not yet ready for formal presentation.

Finally, the use of neural pathway strength analysis provides additional tools for insight and understanding of the structure of ANN predictive models and a useful technique for opening up the "black box".

Chapter 4: Generally Applicable ANN Methodologies

This chapter describes techniques employing ensembles of Artificial Neural Networks (ANNs) that can automatically select a subset of relevant inputs from an entire input signal set; together with neural pathway strength visualisation techniques that illuminate this approach. These are proposed as a contribution to machine learning; specifically as an aid to the understanding of neural networks with application in the fields of hydrology and environment (Abrahart et al., 2012). It is hoped they may also have wider applicability across the fields of Pattern Recognition (Bishop, 1995, 2006), Signal Processing (Cochocki, 1993; Lapedes, 1987) and Predictive Modelling (Grayman et al., 2001; He et al., 2011; Liang and Liang, 2006) both for regression and classification models. Regression models involve the prediction of some numeric quantity, such as flood depth or volume or the area of a forest fire. Classifiers involve prediction of a class label, such as “flood” or “no flood” for a given sewer node or “pass” or “fail” for bathing water quality at a beach.

The common thread running through all of the techniques described here is the opening up of the model "black box" through analysis and use of ANN weights and biases especially from the viewpoint of neural pathways from inputs to outputs of feedforward networks.

The first of these is described in section 4.1. It is an approach to analysis of the overall net effect of each input on each output of an ANN and is designated here as Combined Neural Pathway Strength Analysis (CNPSA).

The second technique, described in section 4.2, involves the creation of an ensemble of ANN models based on division of datasets into a number of folds each containing a number of observation samples and the development of a novel metric for measuring relevance of input features based on variability of neural pathway strengths.

The third involves the use of the above neural pathway strength metric in ensembles of models created as above to determine the relevance of each input "feature" and permit automated feature selection by a meta-modelling

process encapsulating the process that creates the model ensembles. This is referred to here as “Neural Pathway Strength Feature Selection” (NPSFS); see section 0.

The fourth is a visualisation technique (described in section 4.4) for viewing the internal operation of 2-layer feedforward neural networks during training. This reveals the structure of “morphemes” and “sememes” (Hinton, 1984; Hinton et al., 1993) within a 2-dimensional neural pathway strength space and its breakdown into three 2-dimensional subspaces organised by output neuron, hidden neuron or input signal. There are a number of potential benefits to this, including:

- provision of mechanisms for pruning irrelevant connections
- improvement of model performance through such pruning
- provision of a mechanism for deleting hidden neurons
- provision of an alternative mechanism for pruning irrelevant inputs (equivalent to selection of relevant inputs)
- increasing confidence in neural network models by non-expert practitioners, through providing tools for checking the ways models have made use of the information contained in the training dataset
- provision of an additional mechanism for evaluation of the relative effectiveness of ANN training algorithms.
- backtracing and faultfinding to identify root causes of problems with individual ANN models

4.1 Combined Neural Pathway Strength Analysis (CNPSA)

This section describes the first of two main methods of visualisation and analysis of ANN architecture and weight values, by considering not individual weights but the combined effect of all neural pathways from a given input to a given output via all possible hidden layer neurons. This allows the net effect of a given input feature on each network output to be estimated. This estimation is quantitatively approximate (first-order approximation), due to neglecting the effects of (potentially non-linear) activation functions in the hidden and output layers, but nonetheless is demonstrably useful as an approach. This linear

approximation is not unprecedented either, since it is implicit in Hinton diagrams (Rumelhart and McClelland, 1986b) and the methods employed by Olden and Jackson (2002), which are covered separately in the literature review in chapter 2.

4.1.1 CNPSA methodology

Weights for the hidden layer can be represented by a matrix W_1 of dimension [*n x m*] where *n* is the number of inputs and *m* is the number of neurons on layer 1 (the hidden layer); weights for the output layer can be represented by a matrix W_2 of dimension [*m x p*] where *m* is as above and *p* is the number of neurons on layer 2 (the output layer). The product of the 2 matrices will thus be of dimension [*n x p*] and will provide a set of coefficients representing the strength of the combined neural pathways from each input to each output via all hidden layer neurons (for a fully-connected network). Let us call the product matrix W_{io}:

$$W_{io} = W_1 W_2 \qquad (4.1)$$

where (using standard matrix multiplication):

$$W_{io}^{ij} = \sum_{h=1}^{m} w_1^{ih} w_2^{hj} \qquad (4.2)$$

where: *i* is the index for the *i-th* input (row); *j* is the index for the *j-th* output (column); *h* is the index for the *h-th* hidden unit; w_1^{ih} is the element of W_1 for the *i-th* input and *h-th* hidden unit and w_2^{hj} is the element of W_2 for the *h-th* hidden unit and *j-th* output.

Figure 4.1 illustrates (with the blue, curved arrows) two of the 6 possible neural pathways between the third input and the output of a single-output node network. In a 1HL network, there is the same number of possible pathways (between any given input and any given output) as the number of hidden units.

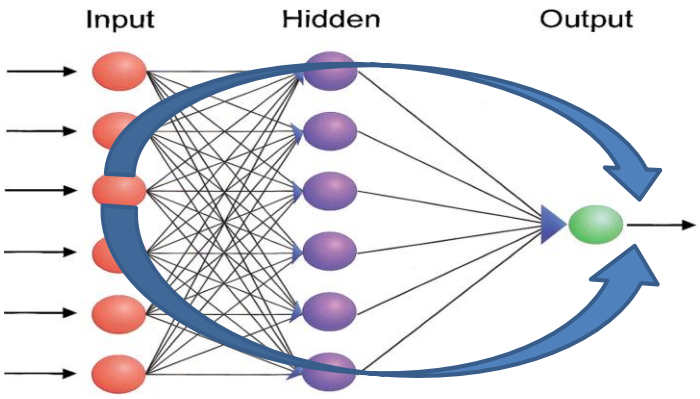


Figure 4.1. Combined neural pathways

Equation 4.3 (discussed fully in Chapter 2 but re-stated here) provides the transfer function for a feedforward ANN. It is worth noting again that this approach neglects the effects of the activation functions κ in equation 4.3, so it is not possible to use W_{io} to compute the output of the ANN, for example based on a vector of input values. Nonetheless the method does provide insight into the relative strengths of influences of each (normalised) input on each output and, as shown below, provides ways of visualising the emergent structure of ANN models during and on completion of training.

$$y = f(x) = \kappa\left(\sum_i w_i g_i(x) + b\right) \tag{4.3}$$

where: x is the input, $g_i(x)$ is some function of x, implemented by the neuron(s) towards the input of the network (for the hidden layer $g_i(x)=x$), w_i is a weight associated with input i, b is a time-invariant bias level and κ is an activation function applied to the output of the neuron. An example is provided in Figure 4.8.

Considering the effects of sigmoidal non-linear activation functions on weight values during training, the least effect will be experienced where the output of a neuron's summation process is around the mean point of the sigmoidal curve. Where the summation output begins to drive against the extremities of the sigmoidal curve, this will tend to have the effect of increasing magnitudes of weight updates significantly in order to achieve an equivalent change to the value output by the neuron as a whole, during the process of

error-reduction in training. Therefore this situation would lead to pathway strengths being increased too, rather than minimised[17].

4.1.2 Illustration using ANN urban-drainage flood model

Chapter 3 has covered the application of ANNs to predictive modelling of urban flooding, for example as required for real-time Early Warning Systems (EWS). Also discussed in chapter 3 is the idea of time-lagged inputs to an ANN. Here, such a trained ANN is used to illustrate the principle of CNPSA. The simplified ANN has 10 input nodes (5-each for lagged rainfall intensity and cumulative rainfall), 3 output nodes modelling 3 manholes in the upstream part of the urban drainage network and 3 output nodes modelling 3 manholes in the downstream part of the urban catchment. Figure 4.2 illustrates this specific network architecture. The lags (0 to -4 timesteps) are shown against the 5 inputs for each of rainfall intensity and cumulative rainfall. For illustrative purposes, all three possible neural pathways from rainfall intensity input 0 (lag) to upstream output node 1 are emphasised in red.

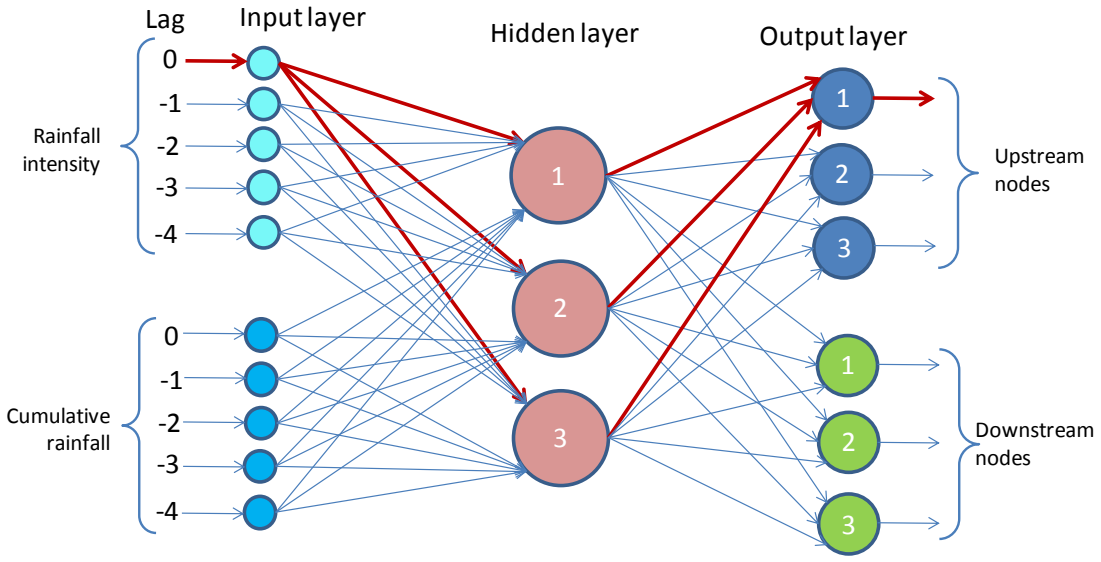


Figure 4.2. Example ANN - emphasising combined pathways from input to output

Figure 4.3 illustrates combined pathway strengths (via the 3 hidden units) between the 3 ANN output nodes that are being used to model upstream sewer nodes (blue bars) and the ANN inputs representing rainfall intensities of time

[17] *An example of a network exhibiting this behaviour is discussed in section 4.4.2.*

lags between 0 and -4 timesteps; similarly for a second set of 3 ANN output nodes modelling downstream sewer nodes (green bars). Because in this illustration there are 2 groups of 3 nodes each, the mean values of neural pathway strengths have been computed for each of the 2 sets, with respect to each input signal. It is convenient to group nodes with similar response characteristics in this way, although by no means necessary. Each output/sewer node could also be treated individually. Figure 4.3 clearly shows that the network is making similar but quantitatively different use of the input signals. However, the ways that each of the 3 hidden units have individually contributed to this are not apparent here[18]. It is also not necessarily the case that all inputs will have a positive (excitatory) effect; it just happens to be the case in this figure.

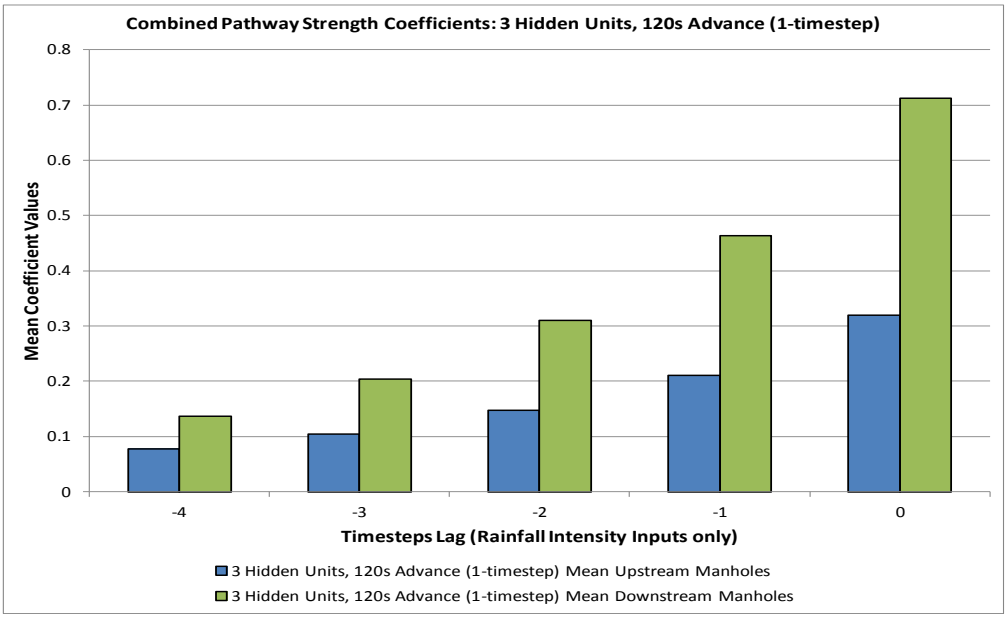


Figure 4.3. Combined pathway strength coefficients for ANN upstream and downstream nodes for rainfall intensity inputs of 0 to -4 timesteps lag

Figure 4.4 similarly shows the combined pathway strengths for the cumulative rainfall input signal. A clear pattern emerges from this analytical approach, which is exploited in section 4.2 when considering the relevance of inputs for an ensemble of models.

[18] *This example has been deliberately chosen with a low number of hidden units, so that a simple pattern of pathway strength coefficients is apparent. Increasing the number of hidden units allows more complex patterns to emerge.*

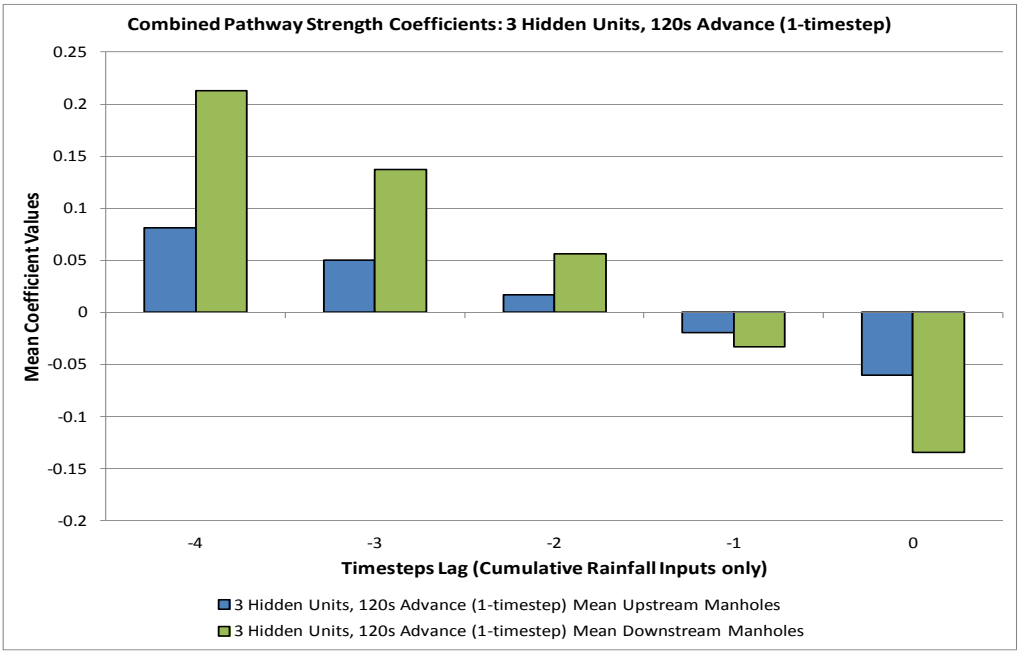


Figure 4.4. Combined pathway strength coefficients for ANN upstream and downstream nodes for cumulative rainfall inputs of 0 to -4 timesteps lag

4.2 NFCV Model Ensemble Generation

Division of datasets into folds is a commonly applied approach in machine learning in order to perform cross-validation of models (Cawley and Talbot, 2003; Hansen and Salamon, 1990; Kohavi, 1995; Tiwari and Chatterjee, 2010a)[19]. It has also been used to create model ensembles. Shen et al. (2012), for example, use K-fold partitioning of a dataset to create an ensemble of models optimised using the Harmony Search algorithm. In hydroinformatics it is often useful to equate the data folds with distinct rainfall "events" and this has been applied extensively in chapter 3 of this thesis. This allows time sequence integrity of the observation samples to be preserved within each event or fold, which is convenient both when using lagged input moving time-windows and also to facilitate graphical inspection of the predicted hydrographs produced by the model output.

[19] *In the limiting case of Leave-One-Out-Cross-Validation (LOOCV), a single sample can be omitted from the training set and used to test the model. This can be repeated using each and every sample in turn as the "left out" test sample. In this way, an ensemble of models with the same number of members as observation samples would be constructed.*

An ensemble of models so produced can be used for prediction including a probabilistic element or predictions from the ensemble can be aggregated in a number of different scenarios including majority voting, worst and/or best case outcomes, maximum-likelihood and estimate of mean. This approach is reported in the literature and reviewed in Chapter 2. A key novel feature (to the author's best knowledge) in this thesis is the analysis of neural pathway strengths (as a function of the layer weights) in the generated ANN ensembles. Combined Neural Pathway Strength Analysis (CNPSA) is described in section 4.1 and this technique is extended to ensembles below.

A method of using Combined Neural Pathway Strength Analysis (CNPSA) results across a whole ensemble of models is sought, so as to investigate its potential for automating selection of relevant ANN inputs and rejection of irrelevant ones. In order to achieve this, a suitable measure for neural pathway strengths is developed.

4.2.1 Ensemble interQuartile Range measure (EQR)

In a NFCV scenario, a set of N similar yet nominally different ANN models are trained; where N is the number of folds into which the training dataset is divided. This is discussed in detail in Chapter 2 as it is a standard technique in the literature for cross-validating results in a variety of machine learning scenarios. Using the CNPSA described in sections 4.1 and 4.1.2 on each member of the ensemble, the aggregate results for the entire ensemble can be presented in the form of a box and whisker plot.

In section 0 this approach is applied to the problem of automation of ANN model input feature selection. The experiment in section 4.3.1 demonstrates the approach using a regression problem related to the environment. The requirement is to predict the area of wildfires in a national park in Portugal based on a set of (coincidentally) 12 input features and a dataset of around 500 observations. This is now used as an example to illustrate the development of the EQR measure.

The example shown in Figure 4.5 presents the spread of combined neural pathway strengths for each of the 12 input features. The set of 12 normalised

input signals is provided to each ANN in the ensemble. The names of the signals are shown on the x-axis (designations are expanded in Table 4.2). Against each input signal is plotted a box-and-whisker, which shows the range of values spanned by the combined neural pathway strengths for that input – across all members of the ensemble. The standard format for each box-and-whisker is used, with the max and min values at the ends of the whiskers, the first quartile at the bottom of the box, the third quartile at the top of the box and the median being the horizontal line across the middle of the box. For completeness, the mean value is also shown (black diamond). It will be noted from inspection of Figure 4.5 that the first 4 input signals (DMC, month, wind, rain) at the left-hand-side are used by more than 75% of the ensemble members in the same sense; whereas for each of the remaining 8 input signals, a significant proportion (*p s.t. 25%<p<75%*) of ensemble members have used it in an inhibitory sense and the remainder have used the input in an excitatory sense[20]. The hypothesis is that those that are used predominantly in the same sense (*p<=25% or p>=75%*) are relevant inputs and those used in a confused sense are not. This statistical behaviour is emergent from the training of the NFCV ensemble of ANNs, rather than any extraneous computational process. The experiment in section 4.3.1 addresses this hypothesis.

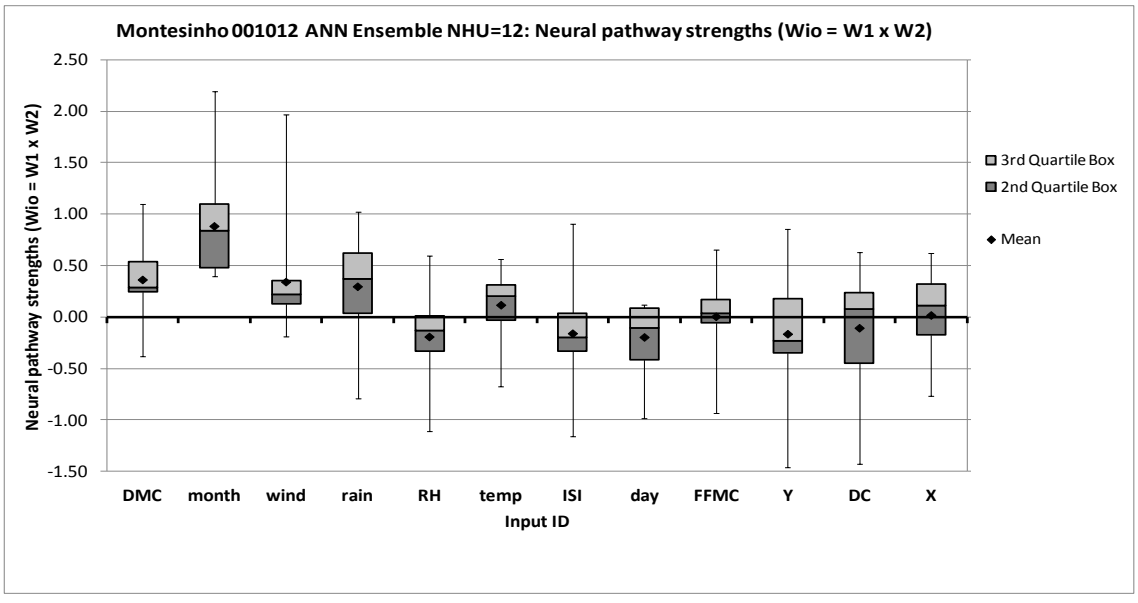


Figure 4.5. Combined neural pathway strength ranges for an ensemble of ANNs

[20] *The box begins to span the x-axis.*

In order to quantify relevance of inputs in this way, a measure, Ensemble interQuartile Range (EQR) is developed. Essentially, this scales the size and distance of the interQuartile box of a box-and-whisker away from the origin (x-axis). Positive values of EQR indicate both Q_1 and Q_3 are on the same side of the origin ("relevant") and negative values indicate a box spans the x-axis ("non-relevant"). The value of EQR also provides a measure of the extent of relevance.

$$EQR = \frac{min\,(|Q_1|, |Q_3|)}{max\,(|Q_1|, |Q_3|)} \cdot sgn(Q_1) \cdot sgn(Q_3) \qquad (4.4)$$

where: Q_1 is the combined neural pathway strength (CNPS) value of the first quartile (bottom of box); Q_3 is the CNPS value of the third quartile (top of box); $|x|$ is the absolute value of x; $sgn(x)$ is the signum function of x s.t.

$$sgn(x) = \begin{cases} -1 & if\ x < 0, \\ 0 & if\ x = 0, \\ +1 & if\ x > 0. \end{cases} \qquad (4.5)$$

Thus EQR exists in the range [-1, 1] $\epsilon\ \mathbb{R}$ for each input signal used for an ensemble of models. Figure 4.5 illustrates the values of EQR for a set of 12 ANN inputs used for the forest fire area predictive application (see section 4.3.1). In Figure 4.5, the inputs have been ranked on the x-axis in descending order of EQR. Values of CNPS and EQR are shown in Table 4.1 for the same ensemble as in Figure 4.5 against the input identifiers and input variable descriptors. The leftmost 3 input variables can be seen to have EQRs significantly above zero (tops and bottoms of boxes are on the same side of the x-axis). The fourth box (rain) is marginally relevant, since the bottom of the box is located very close to the x-axis. This would equate to a value of EQR of 0.061 (i.e. approximately zero).

Table 4.1. Spreads of combined neural pathway strengths and EQR for 12-inputs to an ANN ensemble

Neural pathway strengths (Wio = W1 x W2)								
Relevance Rank	**Input Descriptor**	**Mean**	**Max**	**Q3**	**Median**	**Q1**	**Min**	**EQR**
1	**DMC**	0.364	1.100	0.536	0.291	0.246	-0.379	**0.458**
2	**month**	0.885	2.197	1.094	0.838	0.479	0.394	**0.438**
3	**wind**	0.343	1.971	0.351	0.218	0.129	-0.193	**0.368**
4	**rain**	0.298	1.022	0.621	0.368	0.038	-0.795	**0.061**
5	**RH**	-0.190	0.593	0.012	-0.134	-0.335	-1.113	**-0.037**
6	**temp**	0.118	0.561	0.312	0.206	-0.030	-0.670	**-0.096**
7	**ISI**	-0.158	0.904	0.039	-0.199	-0.333	-1.158	**-0.118**
8	**day**	-0.194	0.121	0.090	-0.105	-0.419	-0.983	**-0.215**
9	**FFMC**	0.007	0.653	0.174	0.038	-0.055	-0.935	**-0.316**
10	**Y**	-0.163	0.852	0.182	-0.231	-0.346	-1.458	**-0.526**
11	**DC**	-0.105	0.627	0.241	0.077	-0.446	-1.429	**-0.540**
12	**X**	0.019	0.623	0.321	0.115	-0.176	-0.767	**-0.549**

Information can also be derived by whether the box as a whole is above or below the x-axis. Boxes above the x-axis indicate that the input variables are positively correlated with the predicted ANN output, in this case $\log_{10}$(final fire area +1) in the national park in Portugal. These are as expected: Duff moisture code (DMC)(De Groot, 1998) is found to be positively correlated with fire area, since it is defined as negatively correlated with rain and relative humidity and positively correlated with temperature (van Wagner, 1974). Month (coded [1 ... 12]) is found to be positively correlated with fire area, since drying occurs during the spring and summer months and the duff layers do not typically become re-charged fully with moisture until very late in the year. Wind speed is also found to be positively correlated with final fire area, as might be expected due to wind promoting the speed of spread of a fire. Rain is marginally positively correlated with fire area, which is perhaps unexpected, but may be explained due to wind and rain being correlated with each other. The other input features have EQR less than zero, meaning they are hypothetically less relevant. Some of these (e.g. RH and temperature) are correlated with DMC, which the ANN may be using as an alternative source of information in this case. Figure 4.6 shows the range of EQR values by input feature, ranked in descending order of EQR.

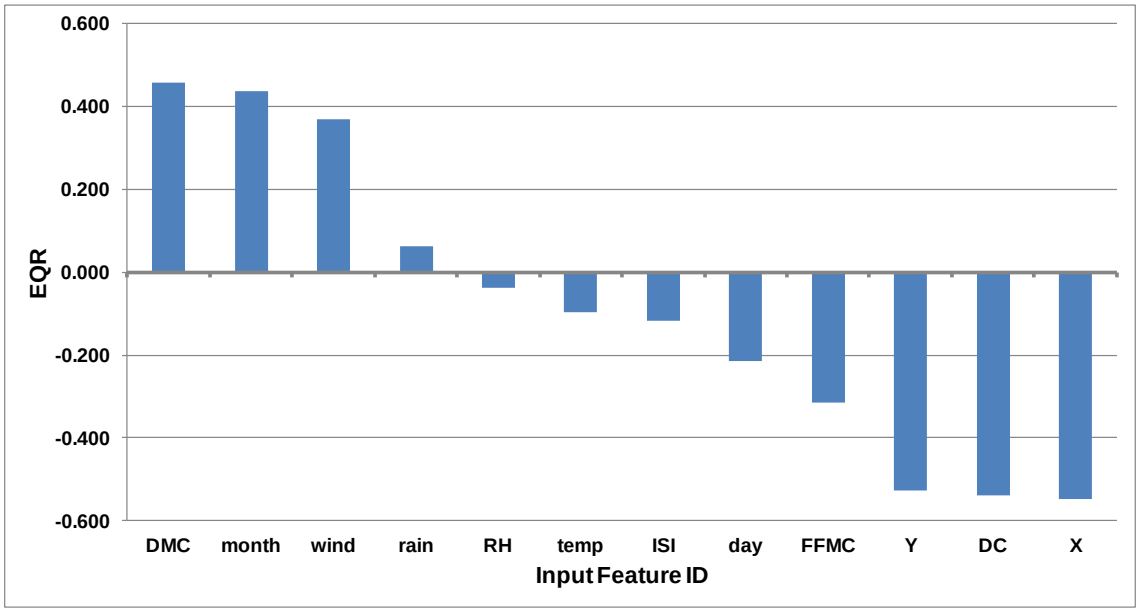


Figure 4.6. Ensemble interQuartile Range (EQR) for 12 ANN ensemble input features

Although the above discussion is not rigorous proof, the availability of physical/environmental explanations of the results learnt by the ensemble of ANN models tends to support the relevance hypothesis and the validity of the EQR measure for input signal relevance.

The pattern of EQR values displayed in Figure 4.6 is found to be fairly typical for this and other case studies using the NFCV approach to ensemble building – and this suggests that EQR is a well-formed measure for input feature relevance. Usually between a third and two-thirds of inputs are measured as being "relevant" using EQR with NFCV ensemble creation.

It would have been possible to base a similar measure to EQR on the maximum and minimum values (ends of whiskers) of neural pathway strengths across the ensemble, but this would be liable to domination by outliers in the ensemble. The approach used is more robust, since it uses the spread of exactly half of the ensemble members occupying the central region of the range of combined neural pathway strengths for each input feature.

4.3 Automated Neural Pathway Strength Feature Selection

The literature review in Chapter 2 covers existing approaches to input feature selection. The objective in all cases is to reduce problem complexity by eliminating unhelpful "non-relevant" input signals or features from the datasets and, as a result, improve model performance and/or computational efficiency. Filter methods pre-process the datasets and evaluate extraneous metrics to determine relevance of input features prior to presentation to the model. Wrapper methods use the model performance itself to evaluate benefits or disbenefits of inclusion/exclusion of input features to/from the datasets. This has to date been largely done by treating the models as black boxes rather than through analysis of the models' calibrated parameters. A notable exception to this has been described in Chapter 2 (Olden and Jackson, 2002), which employs a grey-box approach. The CNPSA/EQR methodology described in this thesis similarly could be described as a grey-box approach.

In contrast to the typical "black-box" approach, the wrapper method presented here extracts and analyses the calibrated ANN weights themselves, learnt from training. These encapsulate what the model has learned about the nature of the problem presented [21]. By applying the CNPSA technique (described in section 4.1) to an ensemble of similar models, the commonality between the models can be analysed using the EQR measure described in section 4.2. The ensemble has been produced here by dividing the dataset using the NFCV approach described in Chapter 2 and in section 4.2, although this is not necessarily a requirement of the method.

The process involves two phases:

- In phase 1, all available input features are applied to all ANNs; and the NFCV method presents different, yet overlapping, subsets of data folds to each ensemble member ANN during training (Figure 2.9). Using EQR, the input features are ranked according to the similarity in their combined neural pathway strengths across all ensemble members.

[21] *The use of early-stopping during training helps to ensure the model's ability to generalise is optimal and to minimise the risk of overfitting to the noise in the training dataset.*

- In phase 2, a feature selection strategy is adopted either by selecting the highest EQR-ranked *n* input features or by choosing a threshold value of EQR, above which to select input features for inclusion in the phase 2 model. The ensemble of models can then be rebuilt and trained using only the selected input features from the dataset. This will have the advantage of reducing computational complexity in any case[22] and the approach can be evaluated in terms of its ability to improve the performance of the models too.

It is hoped that the approach will provide a method for automation of the selection of input features by ANN ensembles directly and thus add to the toolbox of generally applicable techniques available under the heading of Machine Learning.

Section 4.3.1 documents an experiment that demonstrates the approach using a regression problem related to the environment. This well studied problem has been carefully selected to illustrate the general applicability of the method to any predictive, data-driven model. Chapter 5 similarly describes a set of classifier models for bathing water quality prediction. Together, these show EQR feature selection to be effective for both regression and classification.

4.3.1 UCI forest fires dataset case study

This experiment uses a dataset from the University of California, Irvine (UCI) Machine Learning Repository (Bache and Lichman, 2013), a well-known and established library of datasets used extensively for testing machine learning models. The forest fires dataset donated by Cortez and Morais (2008) is used, with the object of prediction of final burnt area due to forest fires in the Montesinho National Park, Portugal, based on a set of 12 input features and a single target feature (the area of fire in hectares). The dataset contains 517 instances. The dataset donors use a support vector machine (SVM) for their solution and achieve best results using a reduct of 4-inputs (temperature,

[22] *The models will have fewer weights in the hidden layer, given the same number of hidden units; a reduction in the number of hidden units may also be possible, due to the simplification of the problem with fewer input nodes/features.*

relative humidity (RH), daily mean wind speed and rainfall total) (Cortez and Morais, 2007). They describe the dataset as follows:

"This is a difficult regression task, where the aim is to predict the burned area of forest fires, in the northeast region of Portugal, by using meteorological and other data."

4.3.1.1 Aims of experiment

1. To build NFCV ensembles of ANN models (using the full input feature set) and evaluate their performance on this "difficult" regression problem
2. To use CNPSA with EQR measure across each ensemble to rank "relevance" of input features
3. To use different neural weight and bias initialisation strategies and network architectures to evaluate the robustness / repeatability (or otherwise) of this feature selection approach
4. To build NFCV ensembles of ANN models (using reducts from the input feature set based on different selection thresholds) and compare their performance with the original ensembles.[23]

4.3.1.2 Methodology

The trial dataset is detailed in Table 4.2, which describes the 12 input features and single target feature, for which 517 instances are available. The values for the target signal are heavily skewed towards zero (fire area), with (otherwise) fire areas spanning 5 orders of magnitude. Therefore (as recommended by Cortez and Morais) the target signal is transformed using $log_{10}(area+1)$ to make the problem a more linear one to solve and render the results more accurate by limiting the dynamic range of output values to approximately [0...3.1]. See Figure 4.7 to obtain an impression of the distribution of the target values in the dataset following log transformation. The target values in the original dataset are also found not to be in truly random sequence, so they are randomly sampled without replacement, before use.

[23] *A mean of input feature rank across all full-feature ensembles evaluated is used for the feature selection, rather than repeat experiments with individual rankings for each ensemble*

The dataset used is as follows:

Table 4.2. UCI Forest Fires Dataset

Index	Input ID	Input Description	Range of values
1	X	x-axis spatial coordinate within the Montesinho	1 to 9
2	Y	y-axis spatial coordinate within the Montesinho	2 to 9
3	month	month of the year	Jan to Dec
4	day	day of the week	Mon to Sun
5	FFMC	Fine Fuel Moisture Code index from the FWI	18.7 to 96.20
6	DMC	Duff Moisture Code index from the FWI system	1.1 to 291.3
7	DC	Drought Code index from the FWI system	7.9 to 860.6
8	ISI	Initial Spread Index from the FWI system	0.0 to 56.10
9	temp	temperature in Celsius degrees	2.2 to 33.30
10	RH	relative humidity in %	15.0 to 100
11	wind	wind speed in km/h	0.40 to 9.40
12	rain	outside rain in mm/m2	0.0 to 6.4
	Target ID	Target Description	
1	area	the burned area of the forest (in ha)	0.00 to 1091

For further information on the Canadian Forest Fire Weather Index (FWI) system and significance of the input features, the reader is referred to WJ DeGroot (1998) and van Wagner (1974).

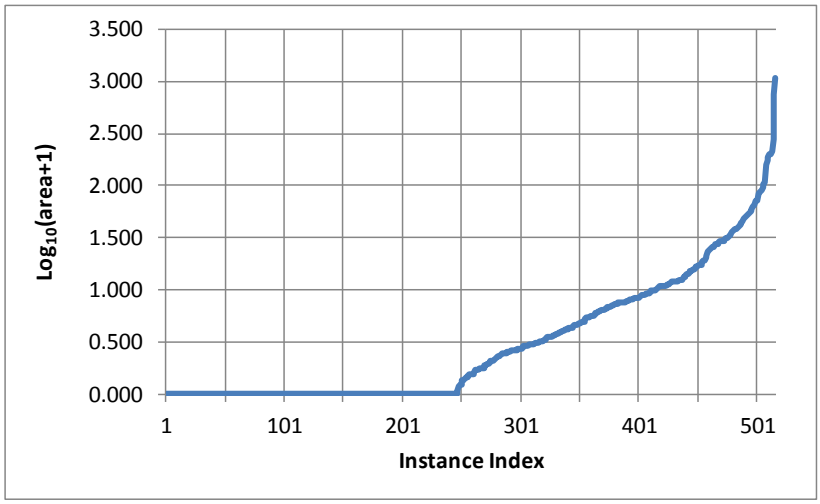


Figure 4.7. Log area target values (sorted in ascending order) versus observation instance

Data preparation

The following data pre-processing and preparation steps are carried out:

1. Take $\log_{10}$(area+1) for the target signal (this is otherwise not normalised)
2. Convert month to integer format [1 .. 12][24]
3. Convert day to integer format [1 .. 7] (Monday = 1)[24]
4. Normalise all input signals to ranges as follows:
    a. [-1, +1]: X, Y, month, day
    b. [0, 1]: FFMC, DMC, DC, ISI, temp, RH, wind, rain
5. Randomise the sequence of instances in the dataset prior to division of the dataset into folds
6. Divide instances into 13 data folds with:
    a. 12 folds each of 36 instances (used for training (10-folds), validation (1-fold) and test (1-fold) of members of NFCV ensemble)
    b. 1 fold of 85 instances (used for final testing of all members of ensemble and excluded from NFCV process)
7. Assign "EventID"s to each data fold [000001 .. 000013]

ANN architecture and configuration setup

The ANN configuration details are as follows:

Feedforward, layered (with 1 hidden layer "1HL"), fully connected, Multi-Layer Perceptron (MLP)(Rumelhart and McClelland, 1986a) ANNs with unlagged inputs and single output neuron are used throughout (since there is only one quantity to predict).

1. Three alternative neural weight / bias initialisation strategies are selected for different ensembles included in the overall experiment (see Table 4.3):
    a. Nguyen-Widrow initialisation method (Nguyen and Widrow, 1990) where all ensemble members are initialised to the same state ["NWS"];

[24] *It would have been possible to use a number of coding schemes for these features, such as 12 or 7 separate binary inputs, but this was not tried, since it would have led to a further increase in number of input features and would not have necessarily furthered the object of the experiment – to demonstrate proof of concept of the feature selection approach.*

b. Nguyen-Widrow initialisation method where all ensemble members are initialised to different states [“NWD”];
c. Uniform randomly-distributed weight and bias initialisation, where all ensemble members are initialised to different states [“UDD”].

2. The Scaled Conjugate Gradients (SCG) (Møller, 1993) optimisation algorithm, using Mean Squared Error (MSE) fitness function is employed for ANN training throughout. This has been done for simplicity and also so as to permit the possibility of initialisation of all ensemble members to the same state. The activation function for all hidden units is hyperbolic tangent (Figure 4.8)
3. Output layer activation function is linear (since this is a regression problem with output span approximately [0 .. 3.1])
4. ANN architectures with 3, 5, 8, 10, 12, 15 and 20 units on the hidden layer are evaluated

Table 4.3. ANN ensemble initialization strategy key / numbers of ensembles in trial

Code	Number of Ensembles	Description
NWS	1	Nguyen-Widrow; all ensemble members initialised to same state
NWD	7	Nguyen-Widrow; all ensemble members initialised to different state
UDS		Uniformly distributed; all ensemble members initialised to same state
UDD	7	Uniformly distributed; all ensemble members initialised to different state

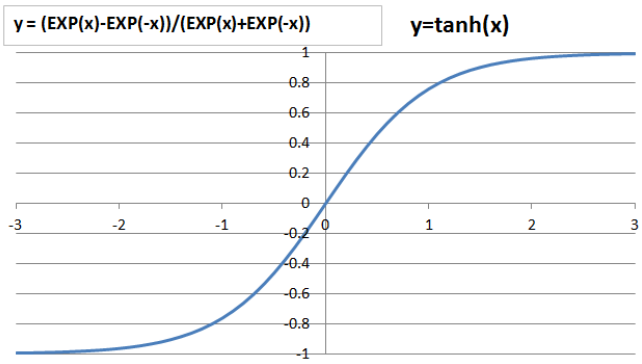


Figure 4.8. Hyperbolic tangent activation function

Early stopping

Early stopping is applied for all trials, by using for each ensemble member a different one of the data folds that is excluded from both the training and test datasets for the ensemble member. This data fold is used for validation of

training progress every 50 epochs during the training. Early stopping occurs if validation error begins to increase by more than 1% over the minimum validation error so far achieved, or validation error stagnates at a fixed level for more than 5 consecutive validation checks. This helps to ensure that over-fitting is avoided.

Ensemble decision-making

The methodology used for ensemble decision making is to assess both the median and mean responses of the ensemble for each sample. The median response tends to be more immune than the mean to the effect of outliers skewing the overall performance of the ensemble. The results for these are compared, to assess which of these strategies is most appropriate.

NFCV ensemble EQR input feature selection trial algorithm

Algorithm 5: NFCV ensemble EQR input feature selection trial

Input: Montesinho forest fires dataset (section 4.3.1.2 "*Data preparation*"); configuration file (section 4.3.1.2 "*ANN architecture and configuration setup*")
Output: set of evaluations of feature selection methodology

1. For each of the above ANN architectures:
2. **Begin**
3. Create a NFCV ensemble of 12 members using the strategy described in section 4.2 and using the first 12 of the 13 data folds described above.
4. For each ensemble member:
5. **Begin**
6. The chosen neural weight / bias initialisation strategy is applied
7. Train for up to 2000 epochs using batch-mode offline training
8. Early stopping is used during training by evaluating ANN validation performance on one of the data folds excluded from the training set for each given ensemble member. Different folds are used for each ensemble member's validation check (each of the 12 folds is used exactly once for validation)
9. On completion of training, simulate with the trained network using the 13th ensemble evaluation data-fold and store responses together with evaluation metrics
10. Store the trained weights and biases and combined neural pathway strength matrix
11. **End;**
12. Evaluate overall NRMSD performance of ensemble using collation of HydroMAT results
13. Evaluate EQR for each input feature using ANaNAS to analyse pathway strength vectors over the ensemble and rank the inputs in descending order of EQR
14. **End;**
15. Assess mean and median rank for each input over all ensembles / ANN architectures / Initialisation strategies
16. Repeat once from 2. using reduct of only the 2 highest median ranked input features
17. Repeat once from 2. using reduct of only the 5 highest median ranked input features
18. Compare NRMSD results for the full 12 input features trial with those for the reduct trials using Student's T-test (Fay and Proschan, 2010)

Limits on ANN architecture sizes relative to training dataset size

Researchers use rules of thumb to determine maximum advisable sizes of ANN architecture, in order to avoid or mitigate problems of overfitting. This is usually expressed in terms of the total number of free parameters (weights and biases) relative to the number of instances (samples) in the training dataset. Maier and Dandy (2000) summarise various authors' recommendations succinctly. The least stringent of these (Rogers and Dowla, 1994) recommends no more than an equal number of free parameters to samples in the training dataset. The most stringent (Amari et al., 1997) recommends a ratio of 30:1 of samples to free parameters.

As shown in section *4.3.1.2 "Data preparation"*, the training dataset for each ANN ensemble member has 36 instances x 10 folds = 360 instances. The number of free parameters in the ANNs used in this experiment is given by:

$$N_{fp} = (N_{in} + 2) \cdot N_h + 1 \qquad (4.6)$$

where: N_{fp} = number of free parameters (weights and biases); N_{in} = number of input features and N_h = number of hidden units. A single output unit is assumed. Figure 4.9 shows the number of ANN weights and biases as a percentage of number (360) of samples in the training dataset for a range of seven architectures with different numbers of neurons on the hidden layer and for three sets of numbers of input features (*NIN* in the key).

This demonstrates that the recommendations of Rogers and Dowla (1994) are met for all ANN architectures used in this experiment. This would give the maximum number of hidden units allowable for the full 12-input feature dataset of 360 samples as 25, following equation (4.6).

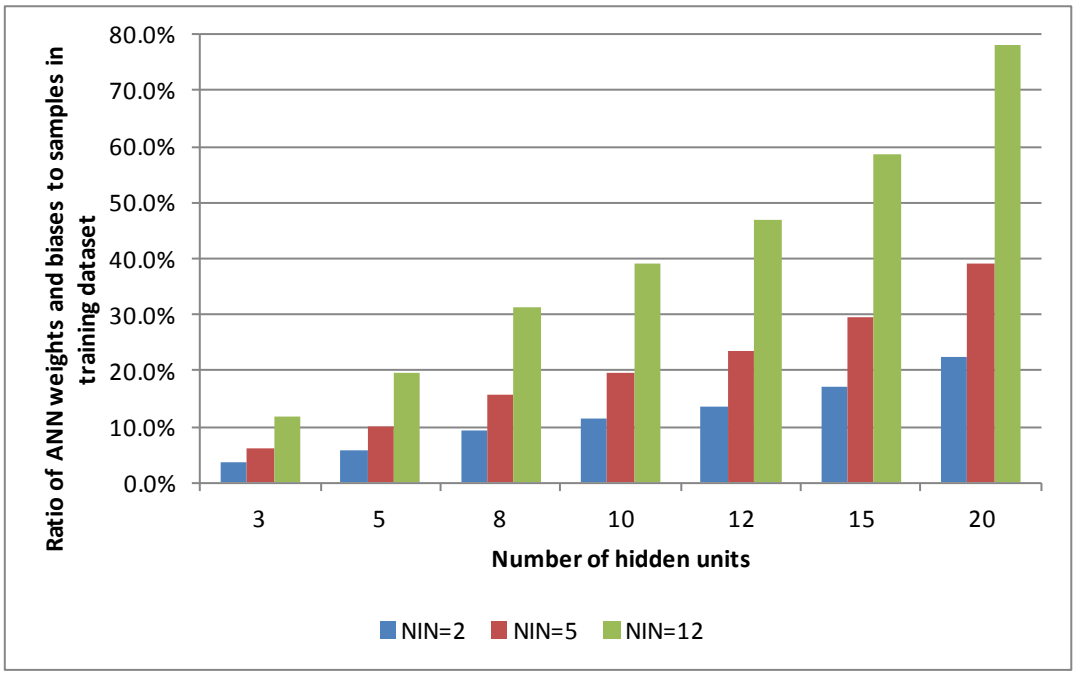


Figure 4.9. Ratio of ANN weights and biases to samples in training dataset as a function of ANN architecture (number of hidden units and input features)

Approach to analysis of performance results for each ensemble

Normalized root-mean-square deviation (NRMSD), comparing $\log_{10}$(area+1) values, is used as a performance metric throughout as defined in the following equation (4.7)

$$NRMSD = \frac{\sqrt{\frac{\sum_{i=1}^{n}(t_i - y_i)^2}{n}}}{(t_{max} - t_{min})} \qquad (4.7)$$

where: n = the number of instances in the dataset; i = index of instance; t = target observations; y = ANN output (predicted values).

1. Using the ensemble test data fold (no. 13), of 85 samples, each ensemble member is presented with the input data and ANN simulated
2. $\log_{10}$(area+1) responses are obtained and compared with the corresponding target responses on a sample-by-sample basis
3. NRMSD results are collated for all ensemble members
4. Additionally, ensemble mean and median responses are also calculated on a sample-by-sample basis for ensemble test data fold 13
5. NRMSD results are also now collated for the ensemble mean and median

Approach to analysis of performance results across ANN ensembles with differing architectures / initialization strategies / reducts of input features

1. The above NRMSD results are tabulated for each ensemble in the given collection of ensembles
2. Box-and-whisker plots provide visual presentation of these results
3. Student's T-tests are used to provide confidence intervals for the null hypothesis when comparing sets of results. Except where paired-sample T-tests are appropriate, unequal variances are assumed, since this gives the most robust T-test result.

Approach to analysis of neural pathway strengths (CNPSA) across collections of ensembles

The methods, described in sections 4.1 (CNPSA) and 4.2 (NFCV) are applied using Algorithm 6.

Algorithm 6: NPSFS across a collection of ensembles

Input: Montesinho forest fires dataset (section 4.3.1.2 "*Data preparation*"); collection of configuration files (section 4.3.1.2 "*ANN architecture and configuration setup*")
Output: set of evaluations of feature selection methodology for collection of ensembles

1. For each of the ensembles in the collection
   (ANN architecture and weight / bias initialisation strategy)
2. **Begin**
3. Create a NFCV ensemble of 12 members using the strategy described in section 4.2 and using the first 12 of the 13 data folds described above.
4. For each ensemble member ANN:
5. **Begin**
6. The chosen neural weight / bias initialisation strategy is applied (Table 4.3)
7. Train for up to 2000 epochs using batch-mode offline training
8. Early stopping is used during training by evaluating ANN validation performance on one of the data folds excluded from the training set for each given ensemble member. Different folds are used for each ensemble member's validation check (each of the 12 folds is used exactly once for validation)
9. On completion of training, simulate with the trained network using the 13th ensemble evaluation data-fold and store responses together with evaluation metrics
10. Store the trained weights and biases and pathway strength matrix (W_{io})
11. **End;**
12. Create box-and-whisker plots of neural pathway strength ranges over the ensemble for each input
13. Evaluate EQR for each input feature to analyse pathway strength vectors over the ensemble and rank the inputs in descending order of EQR
14. **End;**
15. Assess mean and median rank for each input over all ensembles in the collection
16. Produce a scattergram of input feature rankings versus EQR, so as to determine an appropriate decision threshold for selection of input features
17. Calculate mean, median, minimum, maximum, Q1 and Q3 "relevance" rankings for each input feature across the collection of ensembles
18. Produce a box-and-whisker plot of the spreads of these rankings to analyse robustness of

the feature-ranking methodology
19. Compute R-squared correlations of each input signal with respect to the target values and rank input features accordingly[25].
20. Compare median NPSFS input feature rankings across the collection of fifteen ensembles with the R-squared values and display in a bar chart.

Approach to design of repeat experiments using input feature reducts

The trials carried out above result in mean and median rankings of input features over a collection of fifteen ensembles, which use the full dataset of 12 input features.

1. Using the median ranking of the 12 inputs (computed over the collection of fifteen ensembles), the above trials are repeated using reducts of the input feature set with:
   a. The top 2 median-ranked input features for a collection of seven ensembles[26]
   b. The top 5 median-ranked input features for a collection of seven ensembles[26]

Note: The choices of 2 and 5 as the number of input features to use for the reduct testing trials is made following analysis of the results; so is explained fully in section 4.3.1.3.

2. The overall NRMSD performances of the collections of seven reduct ensembles are compared with those of the collection of seven uniformly-distributed all-different (UDD) initialized 12 input-feature ensembles[26]. These are a subset of the fifteen ensembles used in the previously described trials.
3. Student's T-tests are used to evaluate the significance of the results, so as to determine whether the reduct ensembles perform better, similarly or worse than the full input-feature ensembles as well as in comparison with each other.

[25] *This is an approach frequently used in the literature on feature-selection techniques; so it is appropriate to make a comparison with the EQR-based rankings.*

[26] *Each collection of seven ensembles evaluated uses UDD initialization strategy (Table 4.3) and implements an ensemble for each of ANN architectures of 3, 5, 8, 10, 12, 15 and 20 hidden units.*

4. Analysis of the EQR and input ranking performance of the reduct ensemble collections is not repeated, since the method is not intended to be applied iteratively to form ever smaller reducts of the input feature set; but simply to be applied once.

4.3.1.3 Results

Performance results for 12-input feature ensembles

The results in this section relate to methodology sections *"Approach to analysis of performance results for each ensemble"* and *“Approach to analysis of performance results across ANN ensembles with differing architectures / initialization strategies / reducts of input features”.*

Table 4.4 details ANN performance results for seven 12-input ANN ensembles with varying numbers of neurons on the hidden layer (N_{HU} = *[3, 5, 8, 10, 12, 15, 20]*). Each ensemble has 12 members *[ANN01 ... ANN12]* that are initialised using different uniformly random-distributed weight and bias values. Additionally, the NRMSD values taking the mean and median prediction values for the ensemble as a whole are displayed for each observation instance (shaded in blue). These results are also presented in the box-and-whisker plot of Figure 4.10.

A pattern that clearly emerges from this is that the ensemble mean and median performances are significantly better than the individual ensemble members’ NRMSD values. This is demonstrated in the Student’s T-test results in Table 4.5. This shows that the probability of the null hypothesis that the ensemble mean results are from the same population as the individual members’ results is vanishingly small; similarly when comparing ensemble median with ensemble members. Therefore this is at a better than 95% statistical significance level.

Table 4.4. NRMSD values for 7 ensembles with different ANN architectures (12-inputs)

Data fold	000013	000013	000013	000013	000013	000013	000013
ANN Architecture	NHU=3	NHU=5	NHU=8	NHU=10	NHU=12	NHU=15	NHU=20
Individual ensemble members							
ANN01	28.3%	27.2%	26.5%	28.9%	30.9%	27.9%	27.0%
ANN02	26.7%	26.5%	26.9%	27.2%	26.8%	27.6%	28.0%
ANN03	27.1%	26.6%	27.3%	27.1%	27.7%	29.0%	27.0%
ANN04	27.0%	27.1%	27.6%	27.1%	28.3%	28.1%	27.8%
ANN05	27.5%	27.1%	27.8%	26.8%	26.7%	26.8%	27.6%
ANN06	27.4%	27.9%	27.1%	27.7%	28.5%	27.2%	28.3%
ANN07	27.7%	27.1%	27.2%	26.8%	27.0%	27.3%	28.3%
ANN08	27.1%	26.5%	27.4%	27.2%	27.1%	28.7%	27.6%
ANN09	26.6%	26.5%	28.1%	27.2%	26.6%	27.3%	26.7%
ANN10	28.1%	29.3%	28.0%	28.9%	27.9%	29.0%	26.2%
ANN11	27.4%	28.0%	27.6%	27.1%	27.4%	29.5%	28.4%
ANN12	27.6%	27.2%	27.2%	27.8%	27.3%	27.8%	27.2%
Ensemble overall							
MEDIAN	**26.7%**	**26.6%**	**26.7%**	**26.8%**	**26.8%**	**27.1%**	**26.8%**
MEAN	**26.5%**	**26.4%**	**26.8%**	**26.8%**	**27.0%**	**27.1%**	**26.8%**

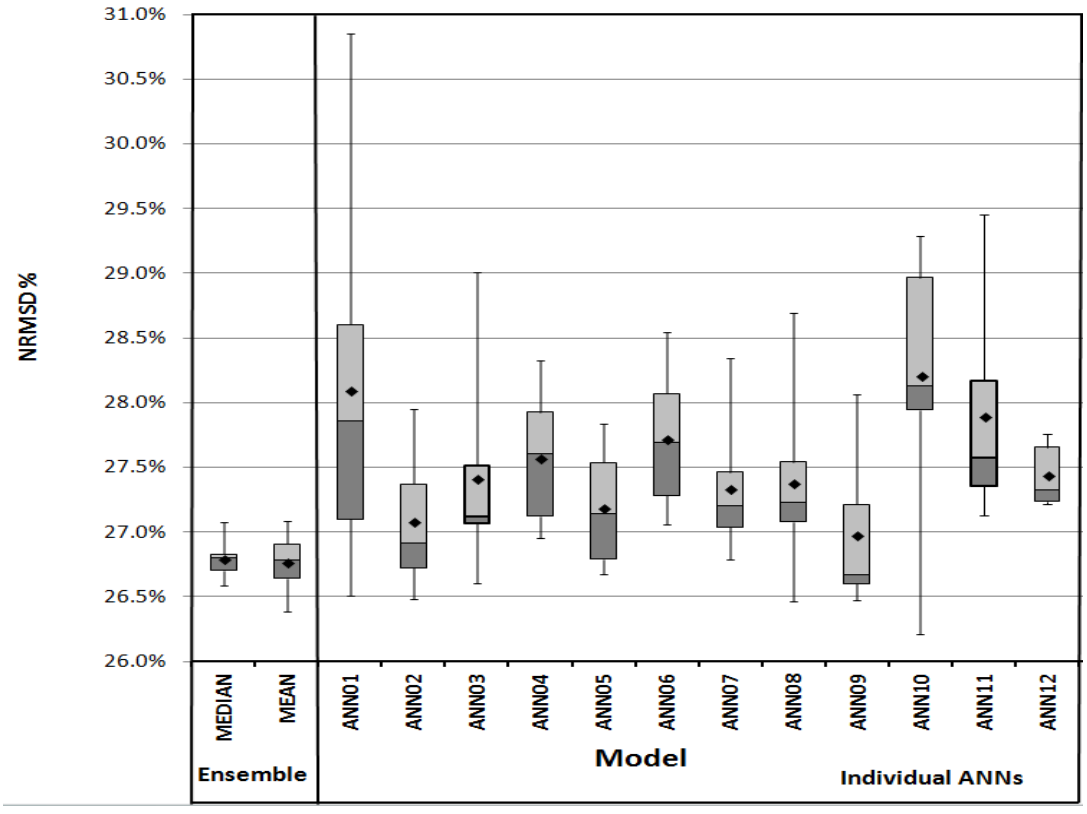


Figure 4.10. Spreads of NRMSD values for collection of 7 ANN ensembles with different numbers of hidden units (12-input) using UCI fires dataset

Table 4.5. Student's T-Test results (assuming unequal variances)

Model	P (1-tailed)	P (2-tailed)
Ensemble Mean : Individual ANNs	3.62E-06	7.24E-06
Ensemble Median : Individual ANNs	4.36E-09	8.72E-09

Combined neural pathway strength analysis (CNPSA) results for 12-input feature ensembles

The results in this section relate to methodology section *4.3.1.2 "Approach to analysis of neural pathway strengths (CNPSA) across sets of ensembles"*.

First, the EQR results are presented for three selected ensembles from the total of fifteen 12-input feature ensembles created during this experiment. These all have 12 hidden units but use three different initialization strategies: [NWS, NWD and UDD]. Table 4.3 contains details.

Table 4.6. Input features ranked by EQR for NWS initialised ensemble

Rank	Input Descriptor	Mean	Max	Q3	Median	Q1	Min	EQR
1	DMC	0.364	1.100	0.536	0.291	0.246	-0.379	0.458
2	month	0.885	2.197	1.094	0.838	0.479	0.394	0.438
3	wind	0.343	1.971	0.351	0.218	0.129	-0.193	0.368
4	rain	0.298	1.022	0.621	0.368	0.038	-0.795	0.061
5	RH	-0.190	0.593	0.012	-0.134	-0.335	-1.113	-0.037
6	temp	0.118	0.561	0.312	0.206	-0.030	-0.670	-0.096
7	ISI	-0.158	0.904	0.039	-0.199	-0.333	-1.158	-0.118
8	day	-0.194	0.121	0.090	-0.105	-0.419	-0.983	-0.215
9	FFMC	0.007	0.653	0.174	0.038	-0.055	-0.935	-0.316
10	Y	-0.163	0.852	0.182	-0.231	-0.346	-1.458	-0.526
11	DC	-0.105	0.627	0.241	0.077	-0.446	-1.429	-0.540
12	X	0.019	0.623	0.321	0.115	-0.176	-0.767	-0.549

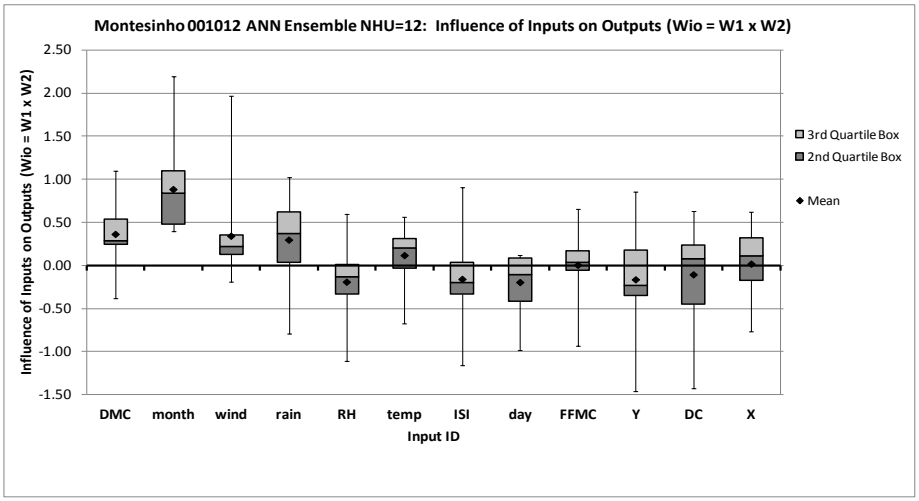


Figure 4.11. Spreads of EQR values versus input feature for NWS initialised ensemble

Table 4.7. Input features ranked by EQR for NWD initialised ensemble

Rank	Input Descriptor	Mean	Max	Q3	Median	Q1	Min	EQR
1	wind	0.525	1.688	0.712	0.426	0.210	-0.090	0.294
2	rain	0.372	1.069	0.690	0.371	0.151	-0.398	0.220
3	month	0.766	1.763	1.287	0.720	0.212	-0.287	0.165
4	ISI	-0.284	0.681	-0.060	-0.243	-0.650	-1.019	0.092
5	RH	-0.235	0.688	0.156	-0.256	-0.530	-1.343	-0.295
6	day	-0.155	0.629	0.144	-0.081	-0.475	-0.818	-0.304
7	Y	0.058	0.571	0.364	0.155	-0.164	-0.728	-0.450
8	FFMC	-0.041	1.108	0.249	-0.107	-0.412	-0.744	-0.605
9	DC	-0.116	0.461	0.337	-0.100	-0.529	-0.859	-0.638
10	X	-0.051	1.074	0.421	-0.086	-0.273	-1.868	-0.647
11	temp	-0.019	0.875	0.193	-0.116	-0.253	-0.790	-0.760
12	DMC	0.158	0.841	0.451	0.387	-0.369	-0.626	-0.819

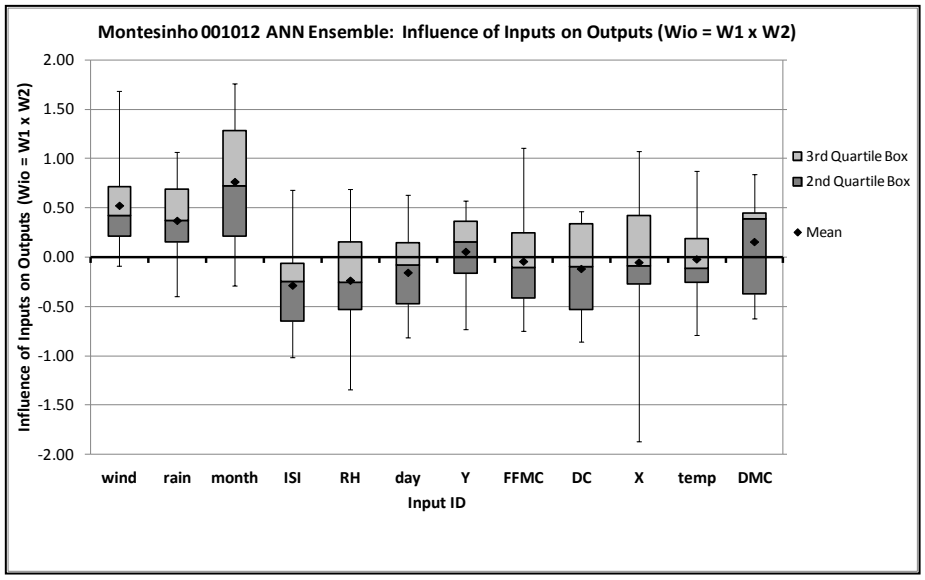


Figure 4.12. Spreads of EQR values versus input feature for NWD initialised ensemble

Table 4.8. Input features ranked by EQR for UDD initialised ensemble

Rank	Input Descriptor	Mean	Max	Q3	Median	Q1	Min	EQR
1	month	0.522	0.955	0.696	0.521	0.361	-0.057	0.518
2	wind	0.201	0.547	0.363	0.302	0.088	-0.353	0.242
3	rain	0.471	1.517	0.665	0.361	0.134	-0.322	0.202
4	Y	-0.325	0.530	-0.027	-0.313	-0.546	-1.338	0.050
5	DMC	0.160	0.718	0.329	0.097	0.002	-0.149	0.005
6	DC	-0.050	0.538	0.139	0.026	-0.329	-0.790	-0.422
7	day	-0.033	0.685	0.092	-0.037	-0.210	-0.531	-0.440
8	X	-0.132	0.219	0.155	-0.076	-0.325	-0.994	-0.479
9	FFMC	0.023	0.628	0.307	0.147	-0.235	-0.769	-0.766
10	ISI	-0.050	0.802	0.332	-0.170	-0.426	-0.725	-0.781
11	temp	-0.003	0.476	0.311	0.184	-0.251	-0.829	-0.809
12	RH	0.030	0.704	0.136	-0.076	-0.145	-0.335	-0.939

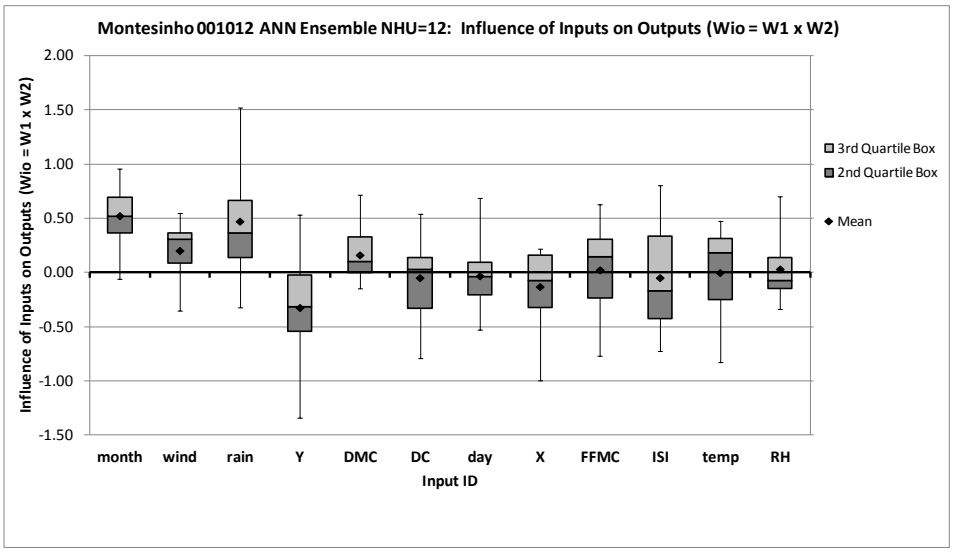


Figure 4.13. Spreads of EQR values versus input feature for UDD initialised ensemble

Table 4.6 and Figure 4.11 show the W_{io} and EQR values for an Nguyen-Widrow (NW) initialised ensemble, with each ensemble member using the same set of initial weight and bias values. Table 4.7 and Figure 4.12 show the same, but using different values of initial weights and biases for each ensemble member. The NW algorithm chooses values in order to distribute the active region of each neuron in each ANN layer approximately evenly across the layer's input space. The input feature rankings for these two ensembles are different, but wind, rain and month feature in the top 4 input features in both cases. As previously discussed, these are plausibly relevant input features. The remaining input features with positive EQRs differ in the case of these two initialisations. In the case of NWS, DMC features as rank 1; whereas it is rank 12 in the NWD initialisation! This is not initially a very convincing result, until we consider the region(s) of the decision space likely to be explored by each of these two initialisation strategies. In the case of the NWS initialisation, all ANNs start their search of decision space from the same point; whereas in the NWD case, each ANN starts from a different locus in decision space. For NWS, it is possible that several ANNs within the ensemble have optimised to the same region of decision space – as evidenced by the small spread of W_{io} values for the box of the DMC input feature. When NWD is used, the height of this box is much greater, indicating that a larger region of decision space (in terms of

weights for the DMC input) is most likely represented within the ensemble. This could be confirmed using visualisation techniques described later in the chapter.

Table 4.8 and Figure 4.13 present the W_{io} and EQR values for uniform distributed all different (UDD) initialisation strategy. Again month, wind and rain feature as three of the top 5, which have EQR>0. Also DMC once again features as one of these; but differently from the NWD strategy. Additionally, Y (north-south coordinates of the centroid of the fire) has shown up as a marginally relevant feature. This is unlikely; as reported by Cortez and Morais (2008), since locations of fires are randomly distributed.

Overall, it can be understood from these results that they are not 100% consistent from ensemble to ensemble and that initialisation strategy may have a significant influence on rankings. The following experiments look at inter-ensemble spread of rankings and EQR values, in order to understand this effect better and to assess the robustness or otherwise of the NPSFS approach to feature selection.

Therefore, the second set of results to be presented in this section summarise input feature rankings and EQR values across ensembles. The neural pathway strength EQR values are calculated for all 12-input features of a collection of fifteen ANN ensembles. These have representative examples using 3 initialisation strategies (see Table 4.3) and with architectures of 3, 5, 8, 10, 12, 15 and 20 hidden units. For each ensemble, “relevance” ranking scores are assigned to each input. A scattergram of EQR versus input relevance ranking is provided in Figure 4.14, for all 12 of the input features used by all ensembles in the collection. Descriptions of the input features are provided in Table 4.2.

Table 4.9 presents the input ranking results for all fifteen 12-input feature ensembles constructed during the experiment. The input features are sorted in ascending order of median rank in this table, with the most “relevant” at the top. The ensembles are arranged in columns. The top two rows in the table define the ANN architecture used (NHU = number of hidden units) and the weight and bias initialization strategy (key in Table 4.3) adopted for each ensemble:

Table 4.9. Input feature rankings for a collection of ANN ensembles

NHU	12	12	3	5	8	10	15	20	3	5	8	10	12	15	20
Init Strategy	NWS	NWD	NWD	NWD	NWD	NWD	NWD	NWD	UDD	UDD	UDD	UDD	UDD	UDD	UDD
Input ID	Rank of input feature by ensemble														
month	2	3	2	3	1	1	1	2	1	1	2	3	1	4	2
wind	3	1	4	1	5	2	2	1	4	2	1	1	2	1	1
rain	4	2	7	2	7	3	3	8	2	5	6	7	3	3	3
ISI	7	4	3	4	4	7	5	9	3	4	12	4	10	6	11
DMC	1	12	8	12	8	4	4	6	6	11	3	11	5	5	7
RH	5	5	1	5	10	10	9	5	5	9	7	2	12	9	9
Y	10	7	12	7	11	12	8	7	9	7	4	6	4	11	5
DC	11	9	6	9	2	8	10	3	12	6	8	9	6	7	6
FFMC	9	8	5	8	3	11	12	4	8	3	10	10	9	8	10
temp	6	11	9	11	12	6	6	12	11	8	5	5	11	10	4
X	12	10	10	10	9	5	7	11	7	12	9	8	8	12	8
day	8	6	11	6	6	9	11	10	10	10	11	12	7	2	12

The scatter cloud of Figure 4.14 intersects the x-axis (*EQR=0*)[27] centred on input feature rank of 5. It is for this reason that it was decided to evaluate a collection of ensembles using a reduct of 5 input features. Figure 4.15 shows the median and spread of rank of input features over the collection of 15 ANN ensembles, whilst also sorting inputs in order of median rank. From this, two input features (month and wind) emerge as clear leaders. It is for this reason that a collection of ensembles using a reduct of these 2 input features is also evaluated.

Since several existing input feature-selection techniques employ R^2 correlation between inputs and observed target signals in the ranking of their relevance, these are computed and the input features are ranked accordingly. A comparison is then made between these and the median rankings produced from the EQR-based collection of fifteen ensembles. Figure 4.16 presents a bar chart of input ranking for both methods against input feature on the x-axis. These are ordered in ascending order of median EQR-based rank for the collection of fifteen ensembles.

[27] *An EQR of exactly zero implies that exactly ¾ (75%) of the ensemble members use the given input feature in the same sense (either excitatory or inhibitory) and the other ¼ (25%) use the input in the opposite sense.*

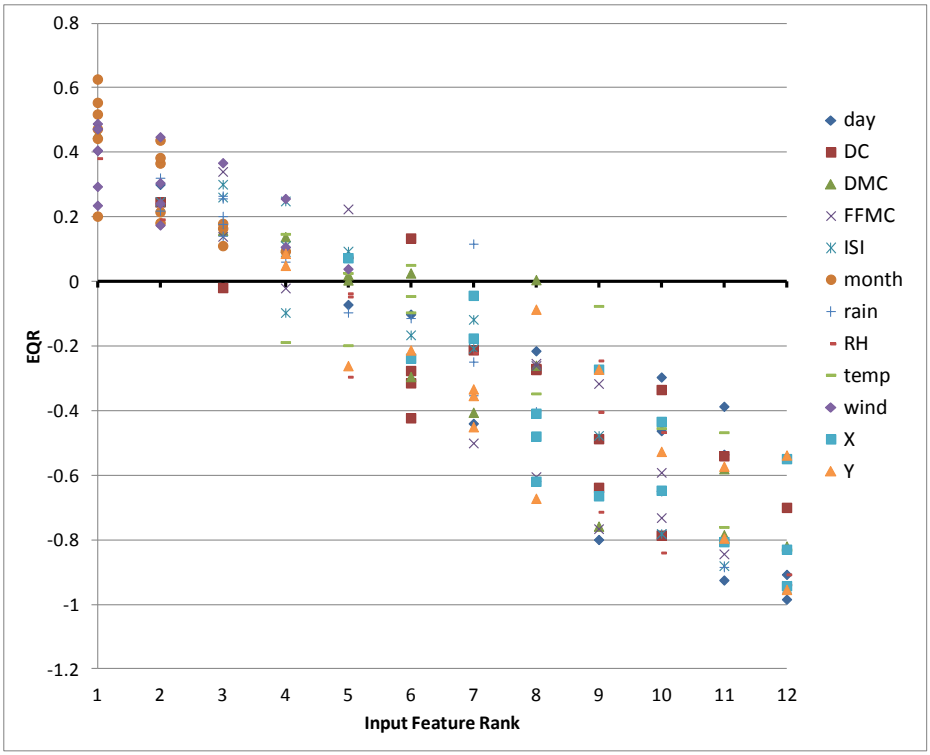


Figure 4.14. EQR versus input feature rank for a collection of 15 ANN ensembles (12-inputs)

The R2 correlation values are very low [0.000 ... 0.013], which is consistent with the donors of the dataset describing this as a "difficult regression problem". Figure 4.16 clearly shows that the R2-based rankings are not closely matched with the EQR-based ones.

Table 4.10. Median rank of input features over collection of fifteen ensembles

Relevance Rank	Input Descriptor	Mean	Max	Q3	Median	Q1	Min
1	**month**	1.933	4.000	2.500	**2.000**	1.000	1.000
2	**wind**	2.067	5.000	2.500	**2.000**	1.000	1.000
3	**rain**	4.333	8.000	6.500	**3.000**	3.000	2.000
4	**ISI**	6.200	12.000	8.000	**5.000**	4.000	3.000
5	**DMC**	6.867	12.000	9.500	**6.000**	4.500	1.000
6	**RH**	6.867	12.000	9.000	**7.000**	5.000	1.000
7	**Y**	8.000	12.000	10.500	**7.000**	6.500	4.000
8	**DC**	7.467	12.000	9.000	**8.000**	6.000	2.000
9	**FFMC**	7.867	12.000	10.000	**8.000**	6.500	3.000
10	**temp**	8.467	12.000	11.000	**9.000**	6.000	4.000
11	**X**	9.200	12.000	10.500	**9.000**	8.000	5.000
12	**day**	8.733	12.000	11.000	**10.000**	6.500	2.000

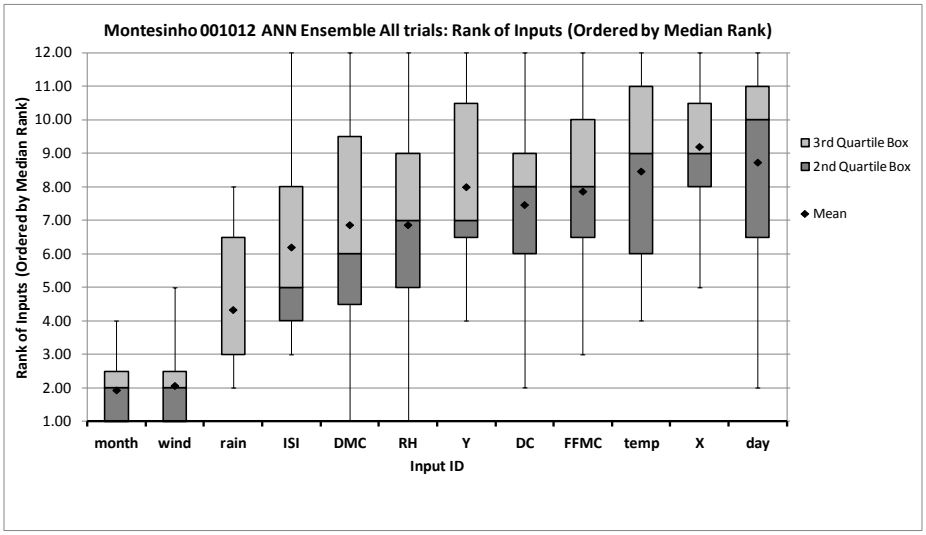


Figure 4.15. Median, mean and spread of rank of input features over collection of 15 ANN ensembles

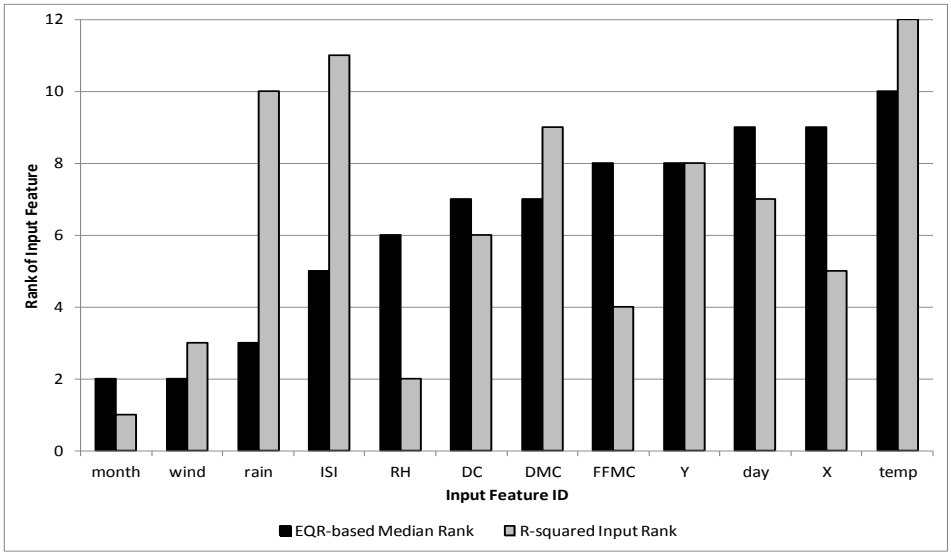


Figure 4.16. Comparison of R^2correlation-based and EQR-based median input feature rankings

Further analysis of the original UCI dataset shows that some input features are negatively correlated with the resultant fire areas. R^2 cannot be used for this, so the Pearson Correlation (Wilcox, 2012) is used in addition. Table 4.11 provides the data for Figure 4.16 and shows that relative humidity (RH)(EQR-rank and Initial Spread Index (ISI) (De Groot, 1998) are the only two input features negatively correlated with fire area. It is reasonable to expect that on days when air humidity is high, combustible material in the Duff Layer (De Groot, 1998) might also have a higher moisture content and therefore be less combustible. The ISI combines the fine fuel moisture code (FFMC) and wind speed to indicate the expected initial rate of fire spread. It is an unexpected

result that this input feature is negatively correlated with ultimate fire area, but the correlation value is in any case very small. However, from the viewpoint of this study, it is clear that the relevances learnt by the ANN ensembles do not closely track the conventional analyses based on R^2 and Pearson correlations between the dependent and independent variables. However, since these latter are very small indeed in almost all cases this only goes to underline the difficulty of this predictive problem and encourages the use of an alternative approach to relevance of inputs.

Table 4.11. R^2 and Pearson correlation values between input features and log(fire area) ordered by EQR-based median input feature ranking

EQR-based Median Rank	R^2 based Input Rank	InputID	R^2	Pearson Correlation
2.00	1.00	month	0.013060	0.114280
2.00	3.00	wind	0.004485	0.066973
3.00	10.00	rain	0.000543	0.023311
5.00	11.00	ISI	0.000107	-0.010347
6.00	2.00	DMC	0.004509	0.067153
7.00	6.00	RH	0.002880	-0.053662
7.00	9.00	Y	0.001508	0.038838
8.00	4.00	DC	0.004404	0.066360
8.00	8.00	FFMC	0.002190	0.046799
9.00	7.00	temp	0.002861	0.053487
9.00	5.00	X	0.003843	0.061995
10.00	12.00	day	0.000000	0.000208

NRMSD performance results from repeat experiments using input feature reducts

The results in this section relate to methodology section 4.3.1.2 "*Approach to design of repeat experiments using input feature reducts*"

Table 4.12 documents the NRMSD performance for three collections each consisting of seven ensembles. Each collection contains ANN ensembles with architectures of 3, 5, 8, 10, 12, 15 and 20 hidden units. Each row in the table represents the spread of combined results for an ensemble of 12 ANNs. The first collection uses the full dataset of 12 input features, whereas the latter two collections are for ensembles using reducts of the input feature set with 5 and 2 inputs respectively (the reader is referred to Table 4.10 for details). Figure 4.17 presents the same data as in Table 4.12 in graphical form. From this is it

possible to gain an impression that the NRMSD error levels for the ensemble collections using the 5 and 2-input reducts are significantly lower than those using the full 12 input features. This is confirmed by performing Student's T-tests on the NRMSD results from the 3 populations of ANNs using 12, 5 and 2 input features.

Table 4.12. NRMSD performance for 3 collections of UDD initialised ensembles; grouped by number of input features used

Inputs used	NHU	Mean	Max	Q3	Median	Q1	Min
12	3	27.4%	28.3%	27.6%	27.4%	27.0%	26.6%
12	5	27.2%	29.3%	27.4%	27.1%	26.6%	26.5%
12	8	27.4%	28.1%	27.7%	27.4%	27.2%	26.5%
12	10	27.5%	28.9%	27.7%	27.2%	27.1%	26.8%
12	12	27.7%	30.9%	28.0%	27.3%	26.9%	26.6%
12	15	28.0%	29.5%	28.8%	27.8%	27.3%	26.8%
12	20	27.5%	28.4%	28.0%	27.6%	27.0%	26.2%
5	3	26.7%	28.7%	26.7%	26.4%	26.4%	26.1%
5	5	27.1%	28.0%	27.3%	27.1%	26.6%	26.3%
5	8	27.0%	28.0%	27.3%	27.0%	26.6%	26.3%
5	10	26.9%	27.8%	27.4%	27.0%	26.5%	26.3%
5	12	26.7%	27.9%	27.0%	26.7%	26.2%	26.0%
5	15	27.1%	29.0%	27.4%	27.0%	26.7%	26.2%
5	20	27.0%	28.1%	27.2%	26.8%	26.6%	26.3%
2	3	26.8%	28.3%	26.8%	26.7%	26.7%	26.5%
2	5	26.8%	27.3%	26.9%	26.8%	26.6%	26.5%
2	8	27.1%	28.9%	27.1%	26.9%	26.7%	26.5%
2	10	26.8%	28.0%	27.0%	26.7%	26.5%	26.0%
2	12	26.9%	27.4%	27.0%	26.9%	26.7%	26.3%
2	15	26.8%	27.5%	27.0%	26.9%	26.6%	26.4%
2	20	27.2%	28.4%	27.5%	27.0%	26.7%	26.4%

The probabilities shown in

Table 4.13 are for the null hypothesis that the compared sets of NRMSD values are from the same population. Probabilities for both 1 and 2-tailed T-tests are presented. In the upper results row, the populations of ANNs using 12 inputs are compared with both populations of ANNs together using the reducts of 5 and 2 input features. Probabilities of the null hypothesis are very low so the likelihood of the input-reduct ANN populations performing better than the full input set ANN populations is significant at the 99% level.

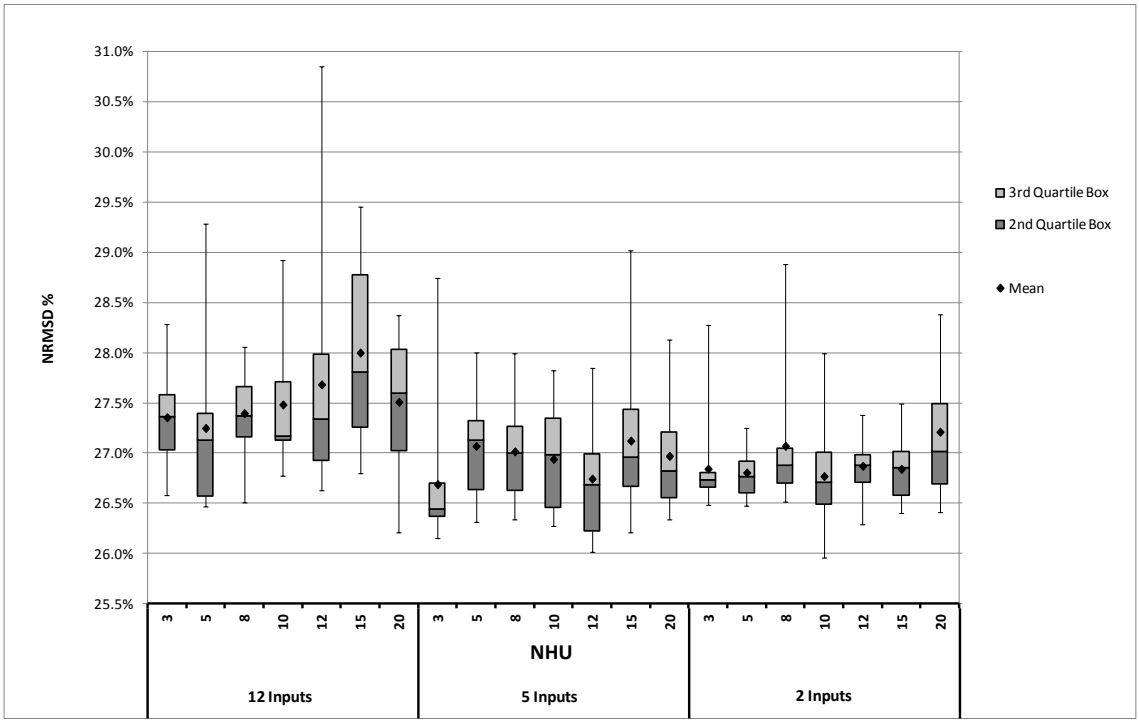


Figure 4.17. Montesinho Predicted Forest Fire Area: Spreads of NRMSD values for 3 collections of 7 ensembles of 12 ANNs

As the populations have significantly different NRMSD performance, it is valid to use the result from the 1-tailed test. In the last row, the NRMSD results from the 2-input and the 5-input reduct populations of ANNs are compared. The probability of the null hypothesis is 0.31; therefore there is no significant difference between the two populations. The 2-tailed test probability applies.

Table 4.13. Student's T-test probabilities comparing NRMSD results for 3 populations of ANNs using different numbers of input features

T-Tests (Two-sample unequal variance)		
Inputs	P (1-tailed)	P (2-tailed)
12 : 2 & 5	0.00007	0.00015
2 : 5	0.15894	0.31789

4.3.1.4 Discussion and analysis

The UCI forest fires regression problem is selected for this case study:

a. in order to demonstrate that the applicability of the NFCV ensemble / EQR input feature selection approach is general; rather than being specific to hydrological models only and
b. because it is a challenging regression problem to model that has an increased likelihood of exposing limitations of the methodology, if any are present.

The problem is to some extent simplified by following the dataset donors' recommendation of taking $\log_{10}$(area+1) rather than area itself as the target signal, reducing the dynamic range of the model output from 5-decades of values to approximately 1-decade. Despite this, overall NRMSD performance of all ANNs was ~26 to 31%; somewhat higher than initially anticipated, but justifiable given the very low R^2 correlations (R^2<0.014) for each input feature with respect to the $\log_{10}$(area+1) target values taken over the entire dataset of 517 instances.

The original study (Cortez and Morais, 2007) from which the UCI fires dataset derived, finds that the best performance is achieved by a Support Vector Machine (SVM)(Hsu et al., 2003) network that uses just four of the 12 inputs (rain, wind, temperature and relative humidity) using the popular Radial Basis Function (RBF) kernel. Their study uses RMSE as the performance metric, so NRMSD can be converted to RMSE using the formula:

$$RMSE = NRMSD\ (t_{max} - t_{min}) \qquad (4.8)$$

The Cortez and Morais reported result of RMSE = 64.7 ha for the best model compares with our result of: NRMSD=26.4%, which converts to RMSE = 10^1.847 = 70.3 ha. This demonstrates that the result for the CNPSA-optimised ANN ensemble is comparable with the best original case study result. Analysis of the sample-by-sample model outputs in comparison with the targets shows that there is a tendency for the models to over-predict the areas for small fires

and under-predict the areas of the largest fires. This is consistent with the use of mean-squared error as the fitness function during training and is well-reported elsewhere in the ANN literature (Garbrecht, 2006; Mukerji et al., 2009). It would be possible to change this model behaviour through the choice or design of other evaluation metrics that weight the importance of (say) samples of larger or smaller area so as to help reduce errors on these. This is not tried here, because absolute model performance is not a central objective of this experiment.

During ANN experiments it is common for a particular ANN architecture to emerge as a clear leader in terms of performance. However, as can be seen from the consistency of the NRMSD results in Table 4.4 no clear leader in terms of number of units on the hidden layer has emerged. On one hand this can be seen as an advantage in that the technique demonstrates insensitivity to ANN architecture but, on the other hand, this may be indicative of insufficient ANN free parameters to be able to model the complexities of the dataset even when using the maximum of 20 hidden units tried. As discussed in section 4.3.1.2 “Limits on ANN architecture sizes relative to training dataset size”, there is a recommended limit of 25 hidden units given the number of samples in the training dataset. The implication may be that more instances would be required in the dataset in order to permit increasing ANN architecture size to the required level of variety for the problem to be modelled effectively.

The EQR measure provides a computationally efficient method of using the ANN training process itself to provide information about input "relevance". The computational cost is $O(N_{fp} \times N_{em} \times N_{in})$ where N_{fp} is the number of free parameters (weights and biases) in each ANN ensemble member; N_{em} is the number of ensemble members; and N_{in} is the number of input signals for which EQR is to be computed. It is worth noting that all these quantities are relatively small. Also the cost is zero-order with respect to the size of the training dataset, since the ANN training needs to take place anyway and this is independent of computation of EQR.

The repeat trials using the reducts of 2 and 5 best-ranked input signals do not include computations for mean and median ensemble performance, since

the first trial NRMSD results are presented in Table 4.4 and Figure 4.10. A pattern that clearly emerges from this, that the ensemble mean and median performances are significantly better than the individual ensemble members' NRMSD values. This is demonstrated in the Student's T-test results in Table 4.5. This shows that the probability of the null hypothesis that the ensemble mean results are from the same population as the individual members' results is not significant ($p<0.01$); similarly when comparing ensemble median with ensemble members. Therefore this is at a better than 99% statistical confidence level.

Table 4.12 and Figure 4.17 for seven UDD-initialised ensembles clearly demonstrate that median and mean performance of a whole ensemble is consistently better than that for individual ANN ensemble members. It is therefore not felt necessary to repeat this result with the reduct ensembles.

4.3.1.5 Conclusions

The experiment demonstrates that it is possible to build NFCV ensembles of ANN models, each member of which performs similarly but not identically to the others. In so doing, it emerges that NRMSD performance of the mean and/or median response of the whole ensemble to each observation (of the full input feature set) is significantly better than that of any of the individual ANN ensemble members. This confirms results from other researchers conducting similar ensemble experiments, referenced in Chapter 2.

When CNPSA with EQR measure is used across each ensemble to rank "relevance" of input features, it is possible to classify inputs as "relevant" or "non-relevant" using a simple discrimination threshold of EQR. As indicated by the scattergram of Figure 4.14, for this problem a reasonable EQR threshold value would be zero, since this would produce a reduct of 5 input features ± 2 from the original set of 12 input features.

By using different neural weight and bias initialisation strategies and network architectures to evaluate the robustness / repeatability (or otherwise) of the feature selection approach, it emerges that the rankings are not perfectly consistent from ensemble-to-ensemble. However, there is a good measure of

repeatability as indicated in Figure 4.15, which shows the spreads of rankings for each input feature across 15 ensembles of ANNs. The selection of inputs for the second half of the experiment is achieved using the median rankings from the entire collection of ensembles. There is arguably sufficient consistency to permit input feature selection to be automated based on an initial trial with a single ensemble. The computational cost of this approach is *O(2N)*, where *N* is the number of members in the initial ensemble (and therefore also in the subsequent reduct ensemble). However, perhaps a more robust approach is to create and train a (small) number of ensembles and use the mean or median ranking from these, as done in this study.

NFCV ensembles of ANN models in this experiment that use reducts of the 2 and 5 highest-ranked features from the full 12-input set are found to perform significantly better than the original 12-input ensembles. This demonstrates that EQR-based input feature selection has worked in this case. Further case studies, using bathing water quality (classification) problems are described in chapter 5, to test the methodology as a generally applicable technique to automate the selection of input features by ANNs for the purpose of model self-improvement for both regression and classification.

Comparing the NFCV ensemble feature selection approach with the Olden and Jackson (2002) randomisation (OJR) approach the following are apparent:

- NFCV ensemble approach provides the advantage of improved performance by taking the median or mean response of the ensemble as a whole; the OJR method has only a single final ANN model.
- NFCV ensemble approach is computationally efficient: it trains a small number of ANNs (12 in this case) on meaningful data and uses all of them; OJR method trains over 1000 ANNs and also needs to perform 999 random permutations of the entire dataset.
- The CNPSA / EQR measure used by the NFCV ensemble is also more computationally efficient than Garson's algorithm or its variant used by OJR, since that requires each hidden unit to be treated separately and 3 measures calculated for each, for all 1000+ ANNs. CNPSA is computed

once for each ensemble member and EQR once for the entire ensemble with respect to each input.

- As a result of the computational load of the OJR method's randomisation trials, it is able to produce a stronger measure of statistical significance of relevance for each input and internal network connection; nonetheless this NFCV ensemble experiment provides a demonstration that sampling the dataset to produce ensemble members (even with only 10% of instances different and 90% overlapping in each case) provides a sufficiently statistically robust result as to be effective in improving NRMSD performance of the reduct ensembles.
- Pruning of internal network connections is not directly achieved with NFCV ensembles (beyond the removal of the hidden weights associated with deleted input features); whereas this is a direct benefit of OJR. An alternative method that could be applied to connection pruning is described in section 4.4.

4.4 Neural Pathway Strength Diagrams (NPSD)

The visualisation technique described in this section is a novel approach in that it treats each pathway through the network (rather than each weight) as an entity and plots it in a 2D space[28]. It aims to provide an additional method of opening up the ANN "black box" and gaining knowledge and insight about the model structure that emerges during the training process and remains on its completion. As will be shown, it can also be used as a diagnostic tool for investigation of problems inside the black box. In order to justify this, existing methods of network weight and pathway visualisation are first examined. A survey of existing methods of visualisation of neural weights and/or pathway strengths is included in the literature review in Chapter 2.

4.4.1 Individual pathway strength analysis

The combined neural pathway strength analysis described in section 4.1 represents combinations of pathways through a neural network from inputs to

[28] *It would be possible to extend the method to 2HL networks by using 3D-plots.*

outputs via all possible paths (each hidden unit as in Figure 4.1). Combined connection strengths from any given input to any given output are represented by signed scalar quantities produced by matrix multiplication of the layer 1 and 2 weights (equation).

It is possible to extend this analysis to look at the strengths of each pathway through a network, from input to output via each possible interconnection. In the case of 1HL networks, this is equivalent to the pathway via each hidden unit. Hinton diagrams provide a means of visualising individual weights in an ANN; an idea about pathway strengths can be inferred by tracing through the Hinton diagram from a given input to an output from row of the hidden layer matrix to column of the output layer matrix, but this is not directly accessible to the eye (see for example Figure 4.18). An alternative way of viewing the network in terms of its transfer function(s) between inputs and outputs is to look directly at strengths of pathways through the network from inputs to outputs. In the case of 2-neuron layer (1HL) networks exactly two weights define the strength of each pathway. It is therefore possible to represent each pathway as a datapoint on a 2-dimensional x-y scattergram in which (as arbitrary convention) the x-axis represents the value of the hidden layer weight and the y-axis represents the value of the output layer weight associated with the pathway. Thus the whole network can be represented on a single scattergram: “Neural Pathway Strength Diagram” (NPSD).

4.4.1.1 Simple ANN example

In order to illustrate the concept, a simple case study ANN for urban flooding is described (similar to those previously discussed in chapter 3). This consists of two output nodes modelling spill depth of water flowing over the weirs of two Combined Sewer overflows (CSOs), three hidden units and two time-lagged input signals (rainfall intensity and cumulative rainfall during each event) each of which has 5 time-lag values associated with it (0 to -4 timesteps) making a total of 10 inputs. In order to train the ANN, a set of 11 design rainfall events (2791 samples) are used and one event (261 samples) is used to validate progress during training and provide early-stopping in the event of validation error starting to increase. The output nodes are chosen such that one

is from a relatively upstream part of the urban drainage network and the other is in a downstream location; that is they display differing amounts of lag between inputs and target outputs.

There now follows a set of pairs of diagrams, the first of each showing the network state prior to training and the second illustrating the state following training. In this case the training took place over 300 epochs using Scaled Conjugate Gradients (SCG) algorithm and Mean Squared Error (MSE) performance metric. For comparison, the Hinton diagrams are presented in Figure 4.18:

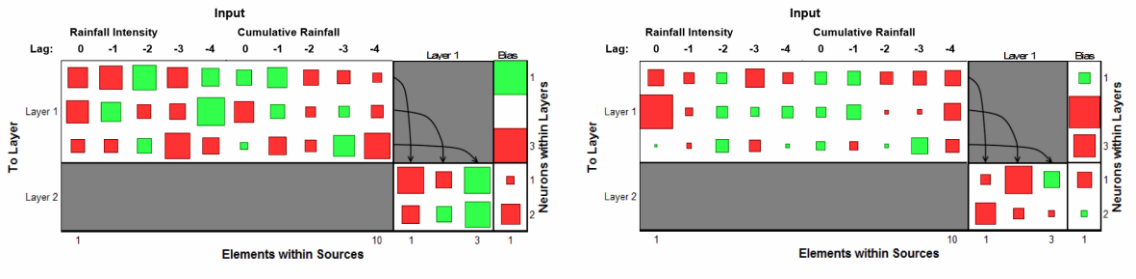


Figure 4.18. Hinton diagram (a) before training (b) after training

The corresponding overall NPSD is illustrated in Figure 4.19 (before training) and Figure 4.20 (after training) for input signals: rainfall intensity (left) and cumulative rainfall (right). The x-axis represents hidden layer weight and y-axis is output layer weight.

Neural pathways are either excitatory (output positively correlated with input) or inhibitory (output negatively correlated with input). In Figure 4.19 quadrants A and C are excitatory pathway regions *A: [+ x +] → +; B: [- x -] → +;* whereas quadrants B and D are inhibitory regions *B: [+ x -] → -; D: [- x +] → -*. It can be seen in Figure 4.19 that all neural pathways comprise a pair of weights that are initialised to within these 4 unit squares.

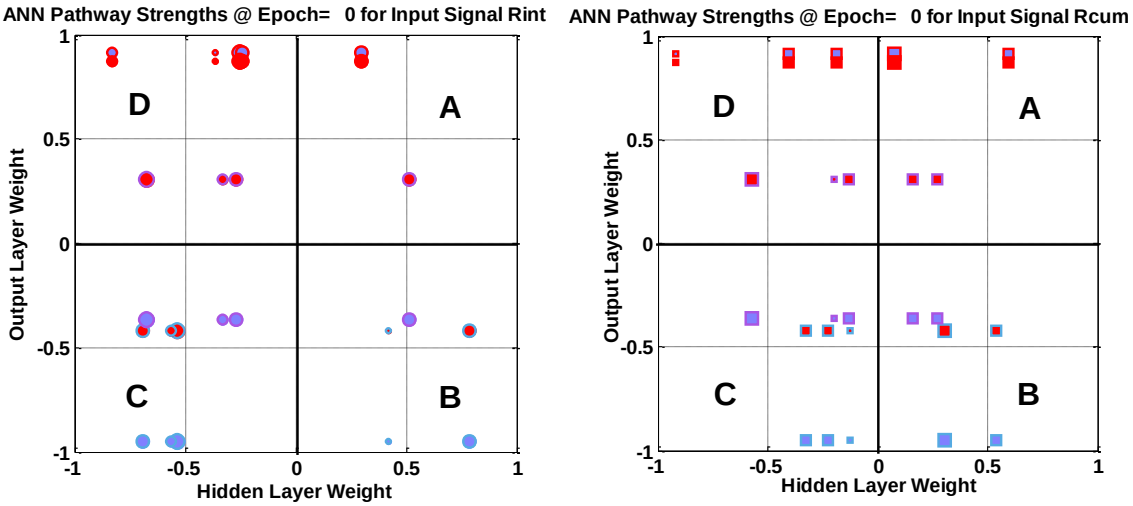


Figure 4.19. Neural Pathway Strength Diagram (before training) for (a) Rainfall Intensity (R_{int}) (b) Cumulative Rainfall Inputs (R_{cum})

In order to reveal structure in the pathway strengths represented in NPSD's the following conventions are adopted:

- Marker shape varies according to input signal
  - here for example, rainfall intensity is represented by circular markers and cumulative rainfall is represented by square markers
- Marker size represents input timestep lag
  - 0 lag is largest, maximum time lag (-4 here) is smallest
- Marker border colour represents hidden unit number
  - first = green → cyan → blue → purple → last = red
- Marker face colour represents output unit number
  - first = green → cyan → blue → purple → last = red

Study of the diagrams reveals that similar symbols are organised in rows of 5, spread horizontally[29]. This is due to there being a single connection (with a single weight value) between an output unit (y-axis) and each hidden unit; whereas each hidden unit has 5 input connections for each input signal, corresponding to the timestep-lags 0 to -4 (x-axis). In Hinton's terms (Rumelhart and McClelland, 1986b) these rows of symbols correspond to "sememes" in that they are units of meaning describing the effect of a time-lagged input signal on a given network output.

[29] *Some symbols overlap, so may not be totally visible.*

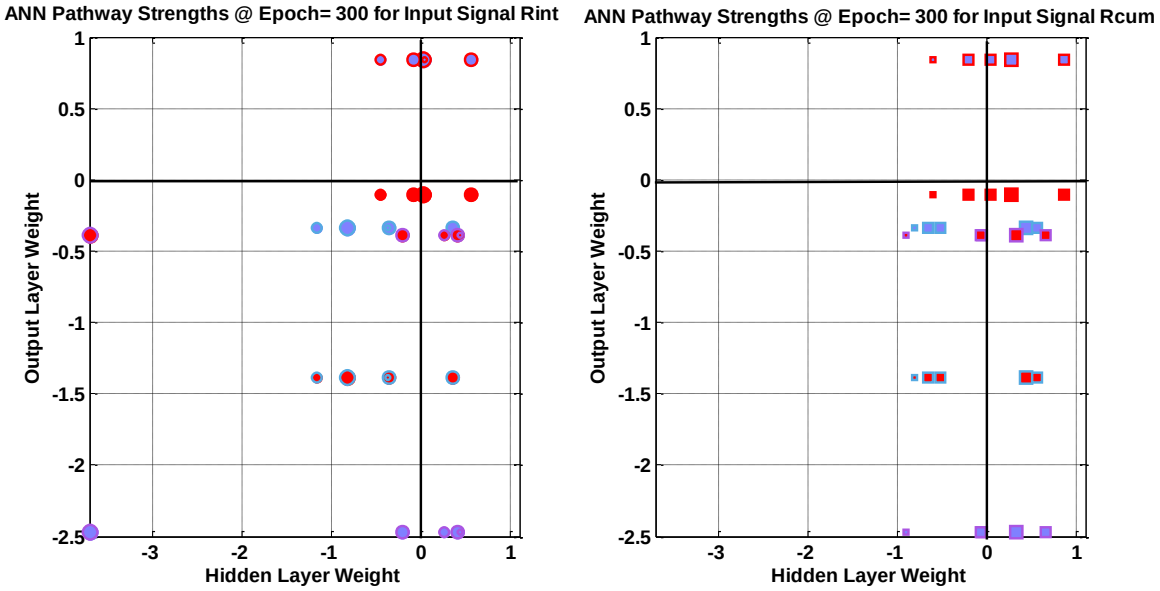


Figure 4.20. Neural Pathway Strength Diagram (after training) for (a) Rainfall Intensity (b) Cumulative Rainfall Inputs

Inspection of Figure 4.19 and Figure 4.20 reveals that there are 6 such sememes in each chart, corresponding to 3 hidden units for each of 2 output units. The face colour of the markers of 3 sememes is red, corresponding with the second (upstream CSO) ANN output node and the remaining 3 sememes have a magenta face colour: the first (downstream CSO) ANN output node. The marker edge colours of the sememes are in 3 pairs [cyan, magenta, red] corresponding to hidden units 1, 2 and 3. Each hidden unit is connected to both output units, which accounts for the pairs of sememes by edge colour. Furthermore, it will be noted that these pairs share the same set of hidden layer weights (x-axis), which can be seen to align vertically between the sememe pairs. Comparison of Figure 4.19 (before training) with Figure 4.20 (after training) reveals that the same colour/size coded sememes exist in both figures; it is possible to infer the direction of travel of each sememe due to the training process. Different x and y-axis scales have been used for the two figures. This experiment has deliberately been set up to be simple so as to reveal clearly the structure of the entire network within a pair of charts. However, in larger networks symbols tend to overlap increasingly. Even so, it is interesting to note that the most influential pathways are the (sparser) ones nearest the edges of the 2-D weight space. For example, the rainfall intensity pathway at [-3.5,-2.5] has a pathway strength of -3.5x-2.5=+8.75, whereas the one directly above it at [-3.5,-0.4] only has a pathway strength of +1.4.

4.4.1.2 Significance of diagram regions

The 2-D weight space described by the NPSD scattergram is located around the origin (0,0), which is functionally equivalent to absence of a network connection. But it can also be seen that pathways on either the x-axis (*,0) or y-axis (0,*) are also equivalent to absence of a connection. A contour plot of pathway strengths over the 2-D weight space is shown in Figure 4.21. Regions of similar colour have approximately equivalent pathway strengths. The left-hand plot (a) is of an extended space with weight values ranging from -10 to +10 in each axis; whereas (b) is the detail of the central region of (a) extending from -1 to +1 in each axis. Colour bar keys to pathway strengths $S_p = w_{ih}w_{ho}$ are provided for each, where w_{ih} = hidden unit (h) weight for input (i) and w_{ho} = output unit (o) weight for its input from hidden unit (h).

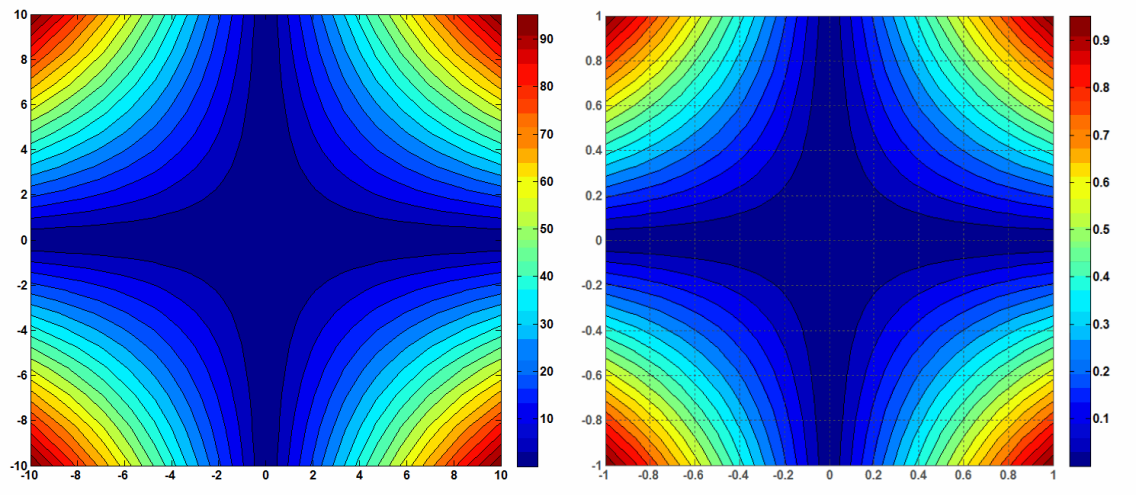


Figure 4.21. Contour Map of Pathway Strengths (a) 20x20 weight space (b) detail of centre 2x2 weight space

This weight space can also be viewed as comprising four quadrants, with quadrants A (+,+) and C (-,-) containing neural pathways with overall excitatory influence from input to output; whilst quadrants D (-,+) and B (+,-) would contain neural pathways with overall inhibitory influence as previously shown in Figure 4.19.

Referring back to Figure 4.20 (following training), we can see that the sememes for both rainfall intensity (left plot) and cumulative rainfall (right plot) for output node 2 (red marker face colour) via hidden unit 3 (red marker border colour) have very low output unit weight values associated with them, yielding

low pathway strengths. It is very likely that the connection between hidden unit 3 and output unit 2 could be pruned with very little effect on the predictive performance of the ANN. Conversely, in Figure 4.20 the sememe for both rainfall intensity (left plot) and for cumulative rainfall (right plot) for output node 1 (blue marker face colour) via hidden unit 2 (magenta marker border colour) have very large negative output unit weight values associated with them, yielding high pathway strengths – especially in the case of the zero-lag input for rainfall intensity (largest circular marker), which has a large negative hidden unit 2 input weight associated with it[30].

Although a simple example has been used here, more complex ANNs may lead to considerable complexity in their NPSDs; therefore it is worth considering a further analysis of the data represented in them to create a number of views as described in the following 3 sub-sections. This arguably provides further insight into larger network structures as will be discussed in the next chapter.

4.4.1.3 Analysis by output node

The NPSDs of Figure 4.19 and Figure 4.20 can be divided into a separate NPSD for each output node. Figure 4.22 illustrates and can be seen to contain the same information as Figure 4.20, organised as a view with respect to output nodes. The left plot is for sewer node CSO SU64003701.3 and the right plot is for CSO SU64004304.2 following completion of training. In general, this view would consist of the same number of sub-plots as output nodes; here 2. Each sub-plot has 3 sememes corresponding to the 3 hidden units in this example.

[30] *Both layer weights being negative produce an overall positive effect on the output with respect to the input – as seen on the combined PSD bar chart in Figure 4.3 for zero lag.*

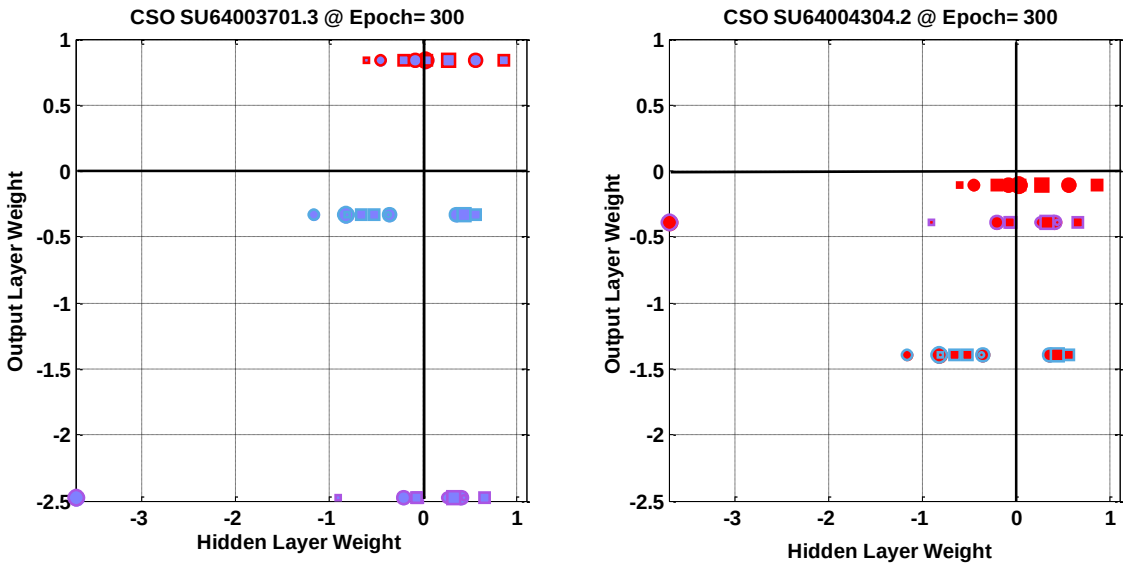


Figure 4.22. NPSD view by output node (a) Downstream CSO node 1 (b) Upstream CSO node 2

The spread of values in each sememe, as before, corresponds with the set of input weights for the given hidden unit – in this case covering both rainfall intensity signal (circular markers) and cumulative rainfall signal (square markers), which are fully connected (all lags) to each hidden unit.

4.4.1.4 Analysis by hidden node

Similarly to the analysis in section 4.4.1.3, the NPSDs of Figure 4.19 and Figure 4.20 can be divided into a separate NPSD for each hidden unit. Figure 4.23 shows 3 views, one for each hidden unit.

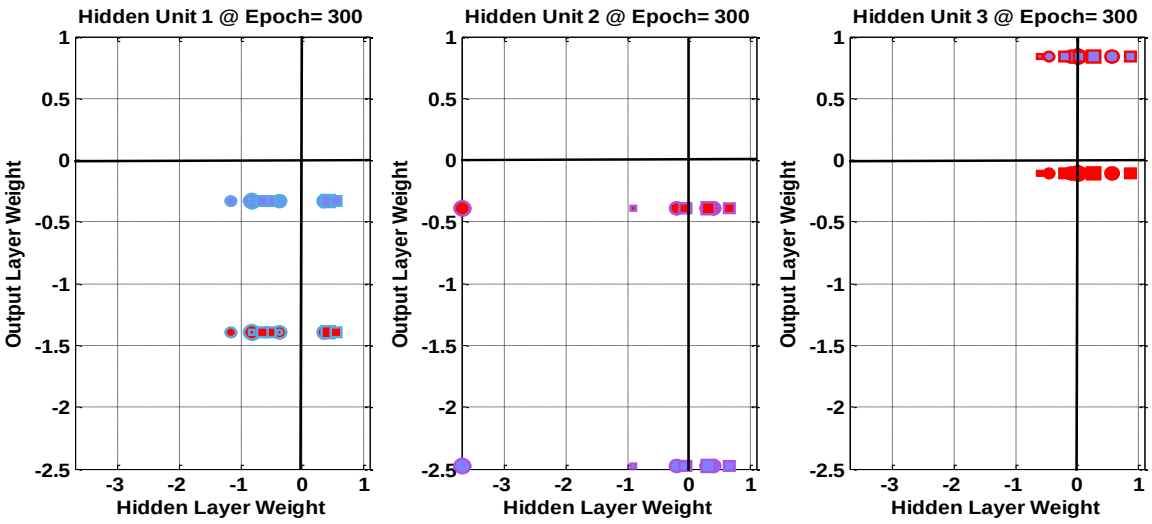


Figure 4.23. NPSD view by Hidden Unit (a) Hidden Unit 1; (b) Hidden Unit 2; (c) Hidden Unit 3

In the plot for each hidden unit, there are 2 rows of markers corresponding with the 2 output units, since each output unit has exactly one connection to each hidden unit and therefore a single weight (shown on the y-axis of the plots) associated with it. The same colour coding scheme as before has been used, meaning that in each plot, the marker border colour is the same for all markers, representing the hidden unit number. The face colour shows which output unit each marker relates to and is therefore the same within each row of markers. Values in the x-axis within each row represent the hidden unit weights for each of the time-lag values of each of the 2 input signals, represented as before by circular and square markers of different sizes according to lag.

From Figure 4.23 (c) it can be seen (for example) that output unit 2 makes very little use of hidden unit 3 at all; whereas Figure 4.23 (a) shows output unit 2 making much stronger use of hidden unit 1. Figure 4.23 (a) and (b) show that both output units use the output of hidden units 1 and 2 in inhibitory (inverting) mode. However, since these sememe rows span quadrants B and C, the overall influence of the inputs is to some extent to cancel each other out.

4.4.1.5 Analysis by input signal

The third and final analytical view is taken with respect to each time lag for both input signals. Recall that in this simple example, time-lag values of between 0 and -4 timesteps have been used. These are illustrated in each of the 5 sub-plots of Figure 4.24, one sub-plot for each lag value. For a hypothetical network, where multiple unlagged inputs are used, this would translate to one sub-plot for each input feature.

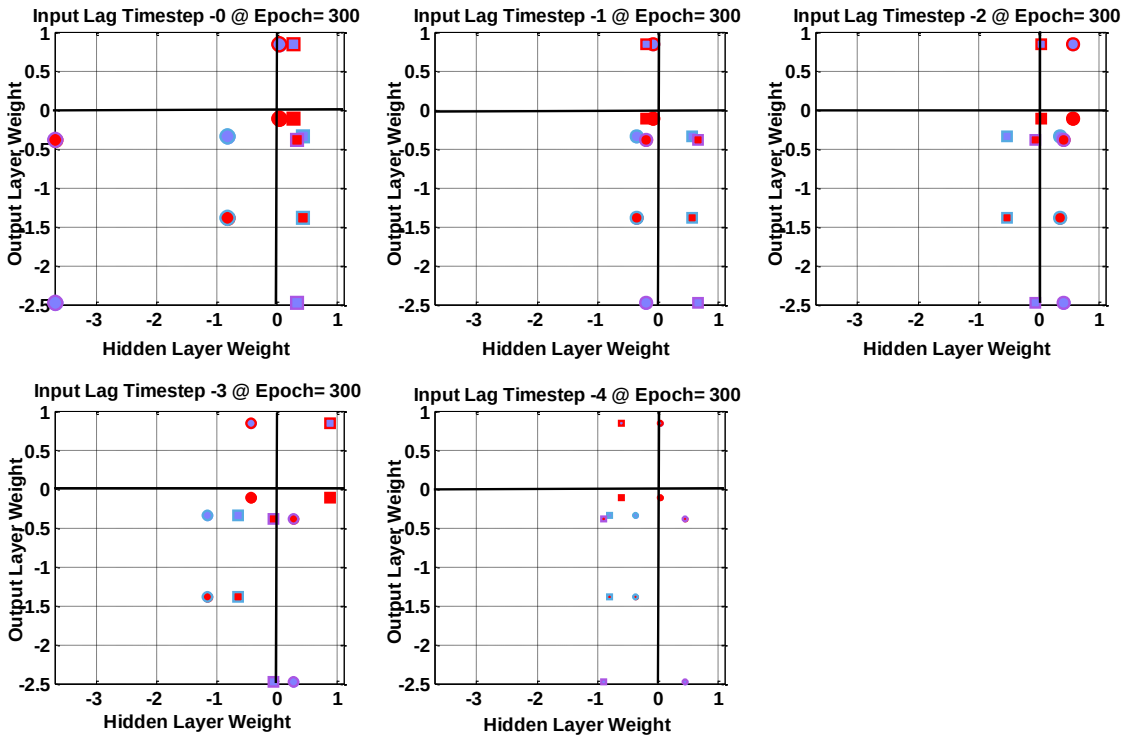


Figure 4.24. NPSD view by Input Lag: Top row: (a) 0 timesteps (present moment); (b) -1 timestep lag (1 timestep previous); (c) -2 timesteps lag; Bottom row: (d) -3 timesteps lag; (e) -4 timesteps lag.

In this view, all marker sizes within each sub-plot are the same, as marker size has been used to represent lag value. It can be seen that markers are organised in pairs aligned vertically. These are a marker for each of the two output units via a single hidden unit and represent the use the output units have made of the specific lag value for a given input signal. Therefore it will be noted that there are 3 pairs of circular markers (rainfall intensity signal) and 3 pairs of square makers (cumulative rainfall signal) representing the pathways through the 3 hidden units. From Figure 4.24 (c) it is possible to see that virtually no use is made of the cumulative rainfall signal (square markers) at timestep lag -2 by hidden units 2 and 3. Only hidden unit 1 uses the cumulative rainfall signal at this lag to a significant extent.

Conversely, it can be noted that a square and a circular marker are aligned horizontally in each of 6 pairs. Each pair represents the different weights (x-axis values) applied to each of the 2 input signals (at the given lag associated with the sub-plot) by a given hidden unit and then weighted by a given output unit (hence a single y-axis value). By comparing (for example) the lower boundary of all 5 sub-plots it is possible to see that output node 1 (mauve

colour marker faces) has applied a large negative weight value to the output from hidden unit 2 (purple marker border colour) and that the strongest pathway is associated with 0 timesteps lag (Figure 4.24 (a)). Note also that the overall effect of this pathway is excitatory with respect to the 0 lag rainfall intensity input, due to the 2 negative weights creating a positive (excitatory) multiplicative product.

4.4.2 NPSDs as diagnostic tools

In the previous section Neural Pathway Strength Diagrams have been shown to provide a set of tools for visualising each neural pathway and the influence it has within a network. Additionally three analytical views of this data, grouped by input signal, hidden unit or output unit allow the structure in the network's weights to be seen more clearly.

In this section the diagnostic capabilities of NPSDs are explored. The UCI forest fires experiment described in section 4.3.1 is used as a case study. The final stage of this experiment involves creation of ensembles of ANNs using reducts of 5 and 2 of the original set of 12 input features – as described in methodology section 4.3.1.2 and results presented in section 4.3.1.3. During the creation of an early trial ensemble with 5-input features and 5-hidden units, one of the twelve ANNs in the ensemble is observed to exhibit a significantly worse NRMSD performance than the other 11 ANNs. Figure 4.25 illustrates the outlier NRMSD performance of ANN02.

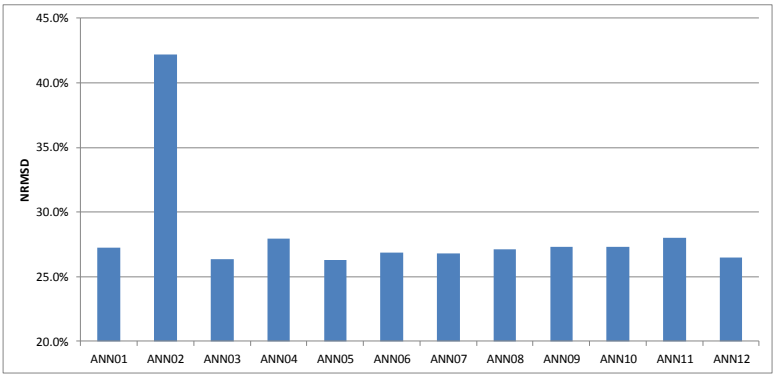


Figure 4.25. NRMSD performance for UCI Forest fires ANN ensemble with N_{in}=5; N_{hu}=5

Further investigation of ANN02's response to each of the 85 sample observations in the ensemble test dataset reveals that its response to a single sample is the cause of this level of NRMSD error.

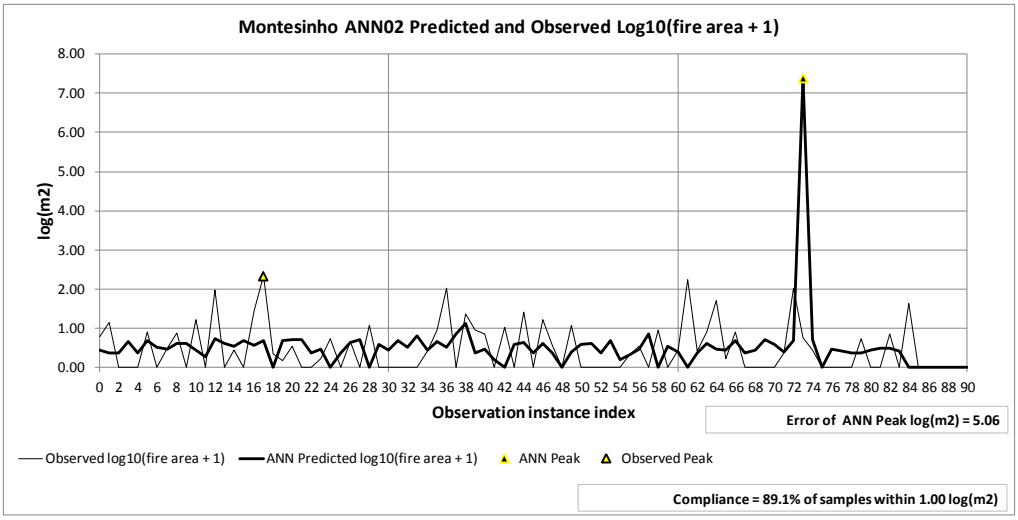


Figure 4.26. Montesinho ANN02 Predicted and Observed Log_{10}(fire area + 1) versus index of observation

Removal of sample 73 from the dataset results in NRMSD for this ANN dropping from 42.2% to 28.9%; much closer to the mean and spread of NRMSD values exhibited by the other 11 ensemble members. However, taking this black-box approach to analysis of performance results tells us almost nothing about the root causes of this emergent behaviour of ANN02.

Using the CNPSA approach described in section 4.1.1 as an analytical tool, it is possible to plot combined neural pathway strengths from each input to the output for each member of the ensemble (Figure 4.27) or by grouping with respect to each input (Figure 4.28).

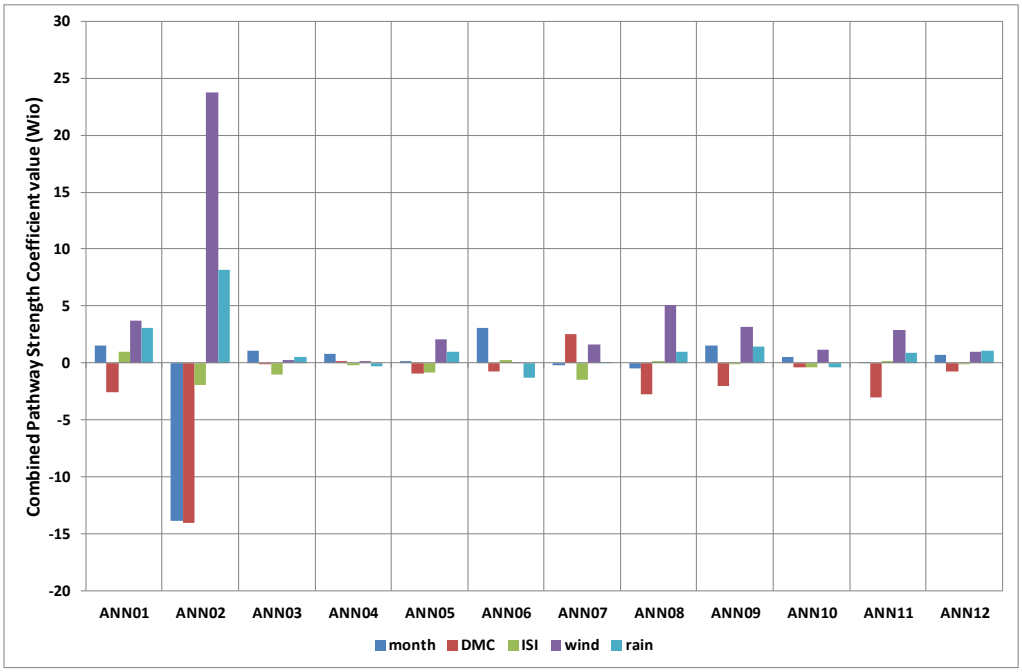


Figure 4.27. CNPSA results for UCI forest fires ANN ensemble with N_{in}=5; N_{hu}=5(grouped by ANN)

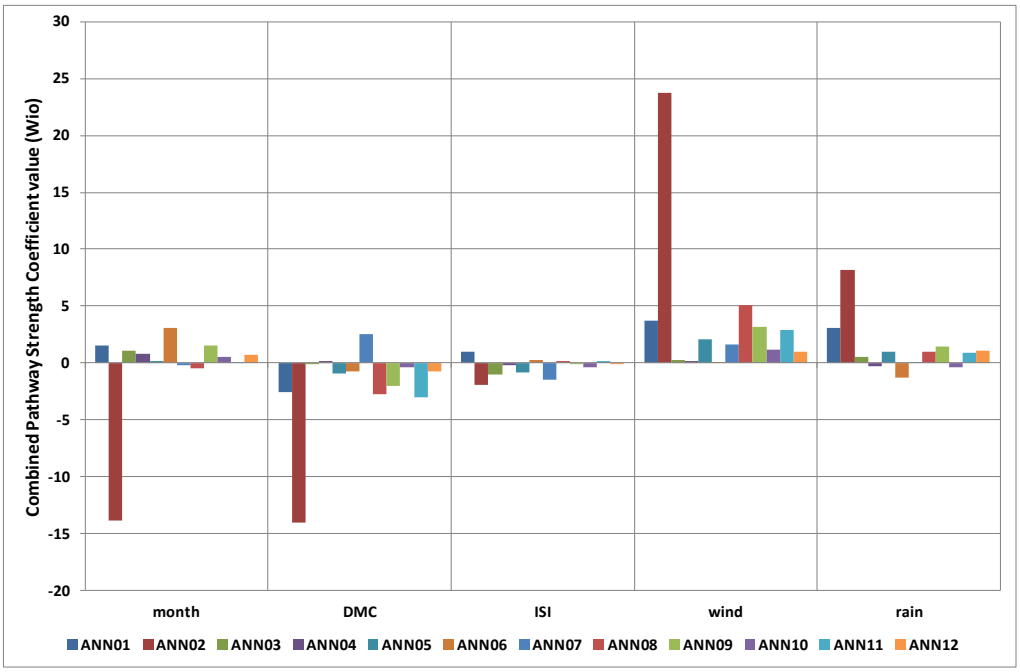


Figure 4.28. CNPSA results for UCI forest fires ANN ensemble with N_{in}=5; N_{hu}=5(grouped by input)

These plots clearly show that ANN02 also has outlying values of combined pathway strengths and that unusually high values (>|10|) occur for three of its inputs (month / DMC / wind). However this does not give a clear impression of the contribution that each hidden unit makes towards this. It is also worth hypothesising that even the ISI input (that exhibits combined pathway strength

of -1.9) may contain individual pathway strengths significantly larger than this, but that they partially cancel each other out.

Recapping the use of a Neural Pathway Strength Diagram (NPSD) to analyse this network shows (Figure 4.29) that this network has 5 input features (shown in the key) and also has 5 hidden units – as can be deduced from the five horizontal rows of symbols, each of which contains one symbol for each input.

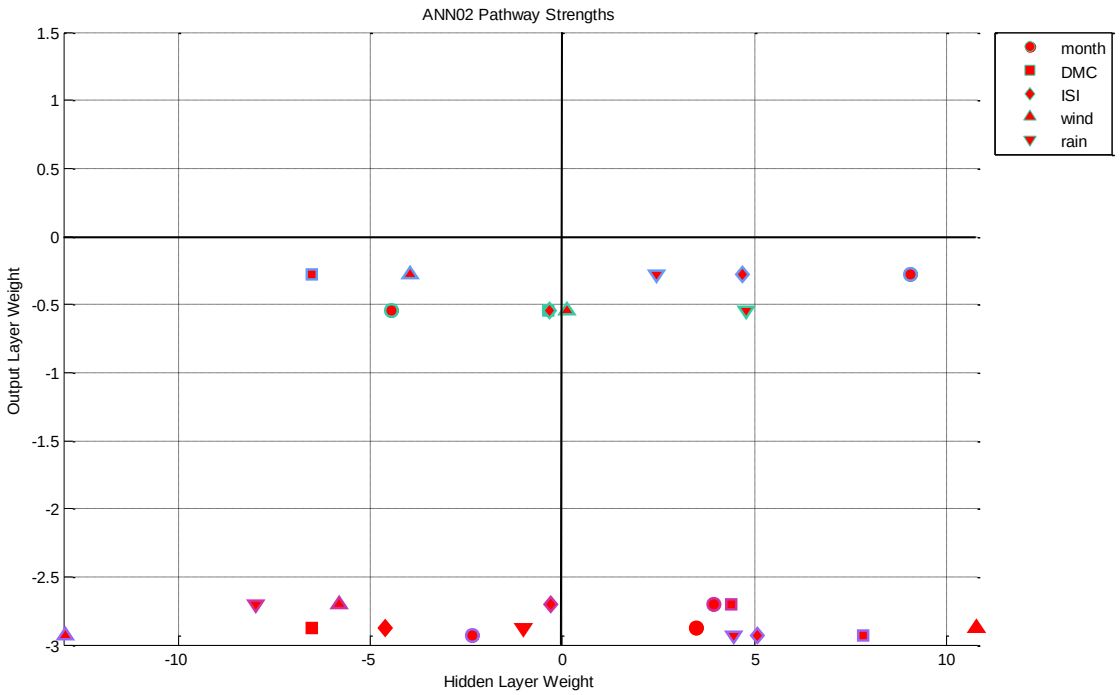


Figure 4.29. NPSD for ANN02

Recall that this network has only one output unit, which accounts for the identical face colour of all the markers. For further details of the significance of the colour coding, please refer to section 4.4.1.1. Before commenting further on these results, it is worth providing an NPSD of one of the other ensemble members as a control (Figure 4.30). This is plotted on the same scale, so that direct comparison can be made. This example (ANN01) is typical of the range of pathway strengths for all other members of this ensemble; though the specific contributions of individual hidden units do vary (as has been widely reported in the literature and can be expected from the symmetry of the ANN architecture and the randomised initialisation).

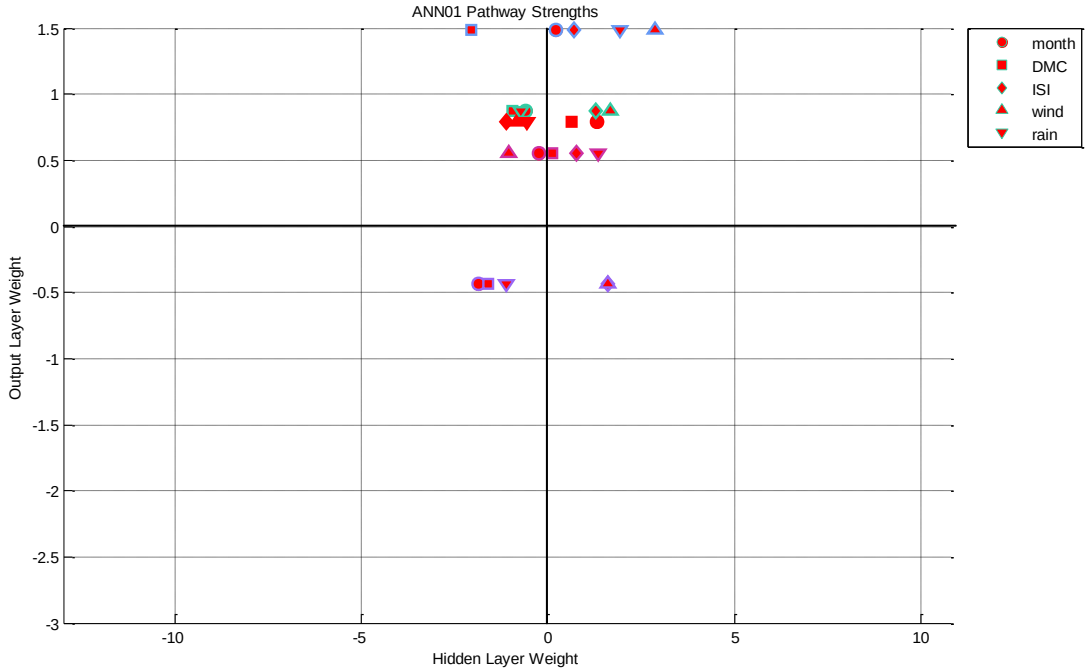


Figure 4.30. NPSD for ANN01

From Figure 4.29 for outlier ANN02 it can be seen that all 5 of the hidden units are contributing to the high pathway strengths with wide extension of input-hidden layer weights w_{ih} (x-axis). However, only 3 of these are also being given a high value of hidden-output weight w_{ho} by the output unit (y-axis). Recall that it is the product $w_{ih}w_{ho}$ that contributes to the summed high-valued combined pathway strengths seen in Figure 4.27 and Figure 4.28. For a further detailed look at the weight structure in this network a breakout analysis of the network by hidden node (as in section 4.4.1.4) is presented in Figure 4.31. This view of the network weight structure clearly shows that it is hidden units 3, 4 and 5 that are making the greatest contribution to the combined pathway strengths, since they are all using hidden-output weight values in the range [-2.5 .. -3.0] (y-axis); whereas hidden units 1 and 2 are only using hidden-output weights in the range of [0 .. -0.5].

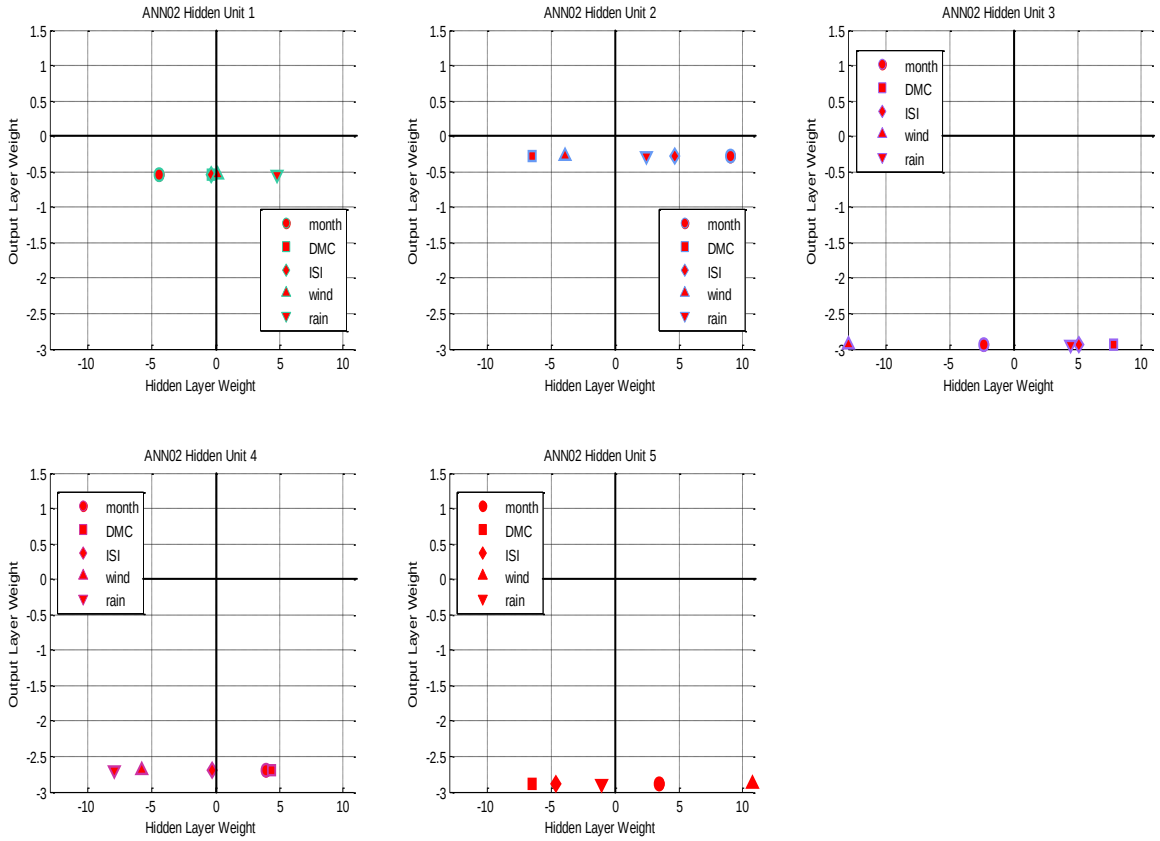


Figure 4.31. ANN02 NPSD view by Hidden Unit: Top row (a) Hidden Unit 1; (b) Hidden Unit 2; (c) Hidden Unit 3; Bottom row (d) Hidden Unit 4; (e) Hidden Unit 5

Furthermore, considering each input by observing the same shaped markers in each sub-plot, it is possible to see that the hidden units are interacting partially to cancel each other out. For example, looking as the ISI input (diamond marker ◆) it can be seen that hidden unit 3 (w_{io} ≈ +5.0 x -3.0 = -15.0) is being counteracted by hidden unit 5 (w_{io} ≈ -4.5 x -2.9 = +13.05). As pathway strengths are additive (neglecting effects of activation functions), this leaves a net effect of only -2.9 for the 2 units combined. So the hypothesis made following Figure 4.27 and Figure 4.28 regarding the ISI input signal's combined pathway strength is indeed found to be supported.

4.4.2.1 Discussion and conclusions on use of NPSDs as a diagnostic tool

From the diagnosis carried out on ANN02 in this ensemble, it is clear that many of the weight values have become set at extraordinarily high magnitudes; especially when compared to other members of the same ensemble. This of course begs the question, "why?".

It is known that the input signals are all in the range [-1 .. +1], since they are normalised as part of the data preparation stage described in section 4.3.1.2. Therefore input-hidden weight values in the approximate range [-13 .. +11] (x-axis of Figure 4.29) will likely result in the maximum value of the data samples driving the outputs of the hidden units' summation functions into the same range of values. As has been discussed, hidden layer activation functions are *tanh(x)* (Figure 4.32) as is established practice. This provides the non-linear regression function-fitting capabilities of the network (Barron, 1993).

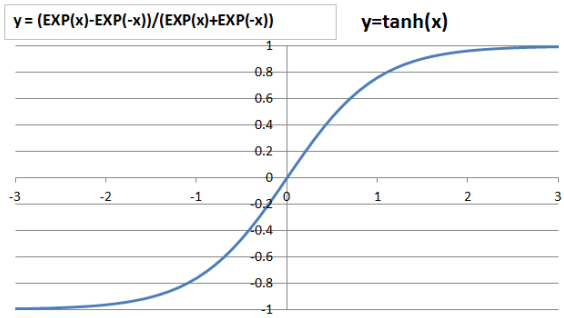


Figure 4.32. Hyperbolic tangent activation function

Input values to the activation functions in each hidden unit are provided by the above-mentioned neural summation outputs. It can be seen from this that input values in the range [-13 .. +11] will drive the activation function outputs far to the extremities of the curve; in the regions very close to +1 or -1, where the gradient of the function is very shallow indeed. This will have the tendency, during training, to cause weight updates to increase in magnitude greatly in order to have any appreciable effect on output and hence on reducing the error being produced by the network. This would therefore imply a tendency for high values of weights in one hidden unit to solicit high values of weights in other units, in order to compensate. In other words, once a training process has begun (based on its initial weights and biases) to explore such a region of the weight space (in at least one dimension) there may be an attractor in other dimensions to explore large values there too. Reintroduction of weight values in the normal range [-1 .. +1] is unlikely to produce sufficient corrective effect against the high-value weights and so the central region of the weight space becomes depopulated. This can be seen for ANN02 in Figure 4.31, with only hidden unit 1 maintaining just 2 pathways within the unit square of hidden and

output weights, out of a total of 25 pathways for the network. ANN01, by contrast, has approximately half of its pathways in this region and all but 2 pathways are in the square region bounded by [(-2,-2);(+2,+2)].

As a diagnostic tool, NPSD visually shows that the problems with ANN02 are of a systemic nature, right across the whole network and are not isolated to a single input, hidden unit or pathway, for example. Merely looking at combined pathway strengths is not sufficient to reveal the full structure in the network connections either; as is shown in relation to the processing of the ISI input as a sufficient example to prove this point.

A suitable solution, used in this case, is to introduce a weight regularisation penalty term into the performance (fitness) function for the training process. This has the effect of penalising large values of weights (and hence pathway strengths) and therefore making exploration of the central unit hypercube of the weight space more attractive. Other solutions may also exist.

In conclusion, NPSDs can be an effective tool to use for diagnosis and visualisation of problems with ANNs, as is exemplified in the treatment of ANN02. They provide insight into the entire profile of an ANN, inasmuch as they do not summarise weight information, but present it all in a form that can be easily assimilated, using the visual recognition capabilities of the human eye.

4.4.3 NPSDs compared with existing visualisation techniques

In Chapter 2, Hinton diagrams, Neural Interpretation Diagrams (NIDs), ARTMAP box plots and Neural Pathway Graphs (NPGs) as existing visualisation techniques are described. Section 4.4.1 introduces the novel approach of Neural Pathway Strength Diagrams (NPSDs) and section 4.4.2 demonstrates an example of their use as a diagnostic tool.

Both Hinton diagrams and NIDs use physical dimension (marker size or line width) to represent weight values so do provide useful tools for visualising complex network structures in a single image. However they lack the numerical quantitative precision of NPSDs in terms of weight values.

The information contained in the physical position of entities in Hinton diagrams and NIDs is also represented in NPSDs, but here it is represented by shape (input feature) and by marker face (output node) and edge (hidden node) colour. To make this more explicit and easier to view, the NPSD breakout analytical views, grouping by input signal, hidden node and by output node are additionally provided.

Therefore it is arguable that NPSDs combine both quantitative accuracy and qualitative representation of information to make them strong rivals of the established visualisation techniques, or at the very least useful additions to the toolbox.

4.5 Conclusion

This chapter investigates and presents techniques proposed as contributions to machine learning suitable generally both for regression and classification models. The following chapter aims to present their application to a further key area of study in Hydrology and the Environment; namely bathing water quality prediction (treated as a binary classification problem).

The techniques proposed include NFCV model ensemble generation and combined neural pathway strength analysis (CNPSA) as an approach to input feature selection for ANN models. An experiment involving building ensembles of ANN regression models to predict final forest fire area based on a number of environmental factors is documented. This demonstrates effectiveness, robustness and repeatability of the approach. Additionally the use of an ensemble of models is demonstrated to lead to improved predictive performance, evaluated using an NRMSD metric.

A further proposed technique, Neural Pathway Strength Diagrams (NPSD), is described in section 4.4. This is a visualisation tool to facilitate visual inspection of ANNs' weights in a way that is quantitatively precise as well as being visually intuitive. Because it does not summarise weight information, but includes all weight values, it is also particularly useful as a diagnostic technique. This is demonstrated using the example of a rogue ANN from the forest fires experiment. Potential further applications of NPSDs would include research into

ANN training, calibration and optimisation algorithms as well as metrics to use as fitness functions in such training scenarios. Additionally, they could be useful for investigation and development of network pruning algorithms.

The common thread running through all of the proposed techniques is the opening up of the model "black box" through analysis and use of ANN weights and biases. These employ the viewpoint of pairs of weights as neural pathways from inputs to outputs of 1HL feedforward networks and show such a viewpoint to be useful in helping to solve machine learning problems such as automated input feature selection, model ensemble generation and problem diagnosis.

Chapter 5: Case study: bathing water quality (Bacti)

This chapter describes case study work carried out as part of an Environment Agency for England (EA-E) funded project with the objective of researching and improving predictive models for water quality at designated bathing beaches in the UK. In the EU, bathing water quality is measured using counts of certain species of bacteria (that are found in the human digestive tract) in sea water samples collected from the bathing beaches. For this reason the project is referred to as "The Bacti Project". Some content of this chapter has been published in IAHR35 Conference, Chengdu, China proceedings as: Duncan et al. (2013d).

5.1 Background

5.1.1 Revised Bathing Water Directive

The revised Bathing Water Directive (rBWD) (2006/7/EC) (European Commission, 2006a) was introduced in 2006 and will take over from the current Bathing Water Directive (76/160/EEC) (European Commission, 1976) in 2015[31]. It sets more stringent water quality standards (see Figure 5.1) and provides a framework for designating beaches as ["excellent" | "good" | "sufficient" | "poor"]. The "poor" designation refers to beaches failing to meet all of the criteria specified in Figure 5.1 and effectively prohibits swimming at that beach. The rBWD also places a strong emphasis on providing information to the public on the quality of bathing waters to allow them to make an informed choice where to bathe. This includes bathing water quality public advice at beaches updated on a daily basis, which generates a requirement for daily bathing water quality predictions at up to 608 currently designated bathing beaches in the UK (DEFRA, 2013). The rBWD also provides a requirement for weekly compliance sampling of water from beaches during the 20-week bathing season from 15 May to 30 September annually. The bacteriological thresholds were developed by the World Health Organisation (WHO) and relate to globally accepted health standards for bathing water quality (Kay et al., 2004).

[31] *This involves a 4-year rolling assessment period for beach designations, which began in 2012 for the 2016 bathing season, to be published late in 2015.*

Heavy rainfall can result in water running off the land, picking up contaminants and overloading the sewerage system, resulting in spillages from Combined Sewer Overflows (CSOs) into receiving waters that may potentially be upstream from bathing beaches. This can quickly have an adverse effect on bathing water quality. Other sources of bacterial pollution can also be triggered by these and other events (Wyer et al., 1997). Mineral sedimentary particulates act as substrates for bacteria growth. These can re-enter suspension in the bathing water and other water from the catchment during or following rainfall or high wind speed events and result in a spike in bacterial count.

	Parameter	Excellent quality	Good quality	Sufficient	Reference methods of analysis
1	Intestinal enterococci (cfu/100 ml)	100 (*)	200 (*)	185 (**)	ISO 7899-1 or ISO 7899-2
2	Escherichia coli (cfu/100 ml)	250 (*)	500 (*)	500 (**)	ISO 9308-3 or ISO 9308-1

(*) Based upon a 95-percentile evaluation.
(**) Based upon a 90-percentile evaluation.

Figure 5.1. Bacterial count criteria for beach designations (European Commission, 2006a)[32]

The Short Term Pollution provision of the rBWD allows for up to fifteen percent of samples taken during such short term pollution events to be discounted from the four year compliance analysis. This is provided that the public is advised[33] in advance that water quality may be unsuitable for bathing, and measures are in place for water quality improvements. This discounting motivates the requirement for predictive modelling.

Where a designated bathing water fails to meet the “Sufficient” standard of the rBWD in 2015, signs will be put up from 2016 advising people not to bathe. This may impact the tourist industry and local economy. Therefore a trade-off exists between public health risks and commercial risks to the tourist industry. Models are thus sought, which are highly accurate and have as low as possible misclassification error rate (“false positives” and “false negatives” as a proportion of total samples). Adopting the convention used inside the

[32] *This covers coastal and transitional waters and is reproduced from the rBWD*
[33] *These are referred to as “Public Advisories” in the results section.*

Environment Agency, false positives are where a bathing water sample exceeds the statutory limits on bacteria count but the model predicts the water to be safe for bathing. False negatives are where a bathing water sample is below the statutory limits on bacteria count but the model predicts the water not to be safe for bathing.

The current Bathing Water Directive (76/160/EEC) (European Commission, 1976) specifies *Faecal Coliform* and *Faecal Streptococci* bacteria counts as the compliance criteria, whereas the rBWD specifies *Escherichia coli (E-coli)* and *Intestinal Enterococci (IE)* respectively as the monitored organisms. These two quality approaches have been shown to be equivalent (Mansilha et al., 2009). This is important, since it means that data gathered since 2000 under the current directive may be used directly for training models for use under the rBWD.

5.1.2 Machine learning model development context

Part of the work described in this chapter forms part of the joint-agency EA-E - University of Exeter Centre for Water Systems (CWS) 'Bacti' project. Novel machine-learning classifier models are developed based on Artificial Neural Networks (ANNs) used to produce Receiver Operating Characteristic (ROC) curves. At the same time, the NPSFS and NPSD methodologies described in Chapter 4 are applied to these models with significant results.

The Environment Agency (EA-E) has developed a set of data-driven modelling tools that predict whether water quality is likely to be above or below a pre-determined bacteriological threshold each day, using as inputs multiple trigger factors from real-time rainfall data and tidal predictions. These models are based on Decision Trees (DT) (Buhrman and De Wolf, 2002; Pal and Mather, 2003; Safavian and Landgrebe, 1991). These are a widely used machine learning technique and involve automatically creating trees of expressions that use the given set of input factors and most closely produce the classifications assigned to each sample in the dataset. Methodology and early results for the DTs included in the Bacti project are described in detail in Tyrrell (2010).

The aim of the Bacti Project is to attempt to improve on the classification accuracy of the DT-based models. So they are included here as a benchmark and comparator for the ANN models described. Misclassifications have inherent risks associated with them; in the case of false positives there are risks to public health; in the case of false negatives there are economic risks to tourism. Therefore these both need to be minimised in the trade-off.

This chapter also summarises recent work on validation of these models and presents a comparison with water quality predictions using a simpler method of single antecedent rainfall triggers that could potentially be applied readily to a larger number of bathing waters. These are also included in the model comparisons.

An existing system known as “BeachLive” (South West Water, 2014) provides live information to the public via website, text messages and Twitter regarding daily water quality at designated bathing beaches in the south west peninsula of England, a popular region for tourism. However, this system relies solely on occurrences of CSO spills as its input. Other factors also influence bathing water quality as is demonstrated in this chapter. It is hoped that this research may contribute to future improvements of systems like BeachLive.

Globally, other researchers employ a variety of modelling techniques for prediction of bathing water quality. These are reviewed in Chapter 2. Particular attention is also given there to other ANN water quality modelling studies and ROC techniques to ensure the optimisation of the operating points of such classifier models.

5.2 Methods of model building and testing

5.2.1 Case study beaches

Models are built and tested for case study bathing waters located at coastal beaches in South West England as indicated in the map of Figure 5.2 and the bathing water unique reference numbers (URNs) and Ordnance Survey (OS) grid references of the beach sample points in Table 5.1.

Table 5.2 contains the corresponding stream sample point OS grid references. Freshwater streams flow across many of the case study beaches and these are a potential transport mechanism for bacterial pollution from a variety of sources, particularly during and following significant rainfall events. Therefore separate samples are gathered from locations near the mouth of streams, upstream from the beach.

Figure 5.2. Case Study Beaches (SW England)

Table 5.1. Case Study Beach Sample Locations

Data start for model	2012 Daily Samples?	Beach Name	Beach URN	OS Grid Beach Easting	OS Grid Beach Northing
2001		LYME REGIS (CHURCH) BEACH	70114403	334430	92126
2000		MOTHECOMBE BEACH	70910108	261050	47340
2000	Yes	SEATON BEACH (CORNWALL)	H1314870	230370	54330
2001	Yes	EAST LOOE BEACH	81414820	225700	53170
2004	Yes	READYMONEY COVE BEACH	81510150	211830	51080
2003	Yes	PAR BEACH	81614842	208510	53140
2000	Yes	PORTHLUNEY BEACH	81814942	197340	41290
2005		ROCK BEACH	82511846	192770	75790
2002		ILFRACOMBE (CAPSTONE) BEACH	73115120	251909	147841
2001		COMBE MARTIN BEACH	73115166	257720	147320
2000		BLUE ANCHOR WEST	E0900700	302161	143509
2005		BURNHAM JETTY	60010410	330235	148684

Table 5.2. Case Study Stream Sample Locations

Stream Name	Stream URN	OS Grid Stream Easting	OS Grid Stream Northing
RIVER LIM AT BEACH (LYME REGIS)	70110104	334223	92129
RIVER ERME AT MOUTH (MOTHECOMBE)	70910110	261350	47300
RIVER SEATON AT SEATON BEACH	81310201	230329	54506
MOUTH OF LOOE ESTUARY - W BANK	81410215	225516	53033
READYMONEY COVE STREAM	81510151	211750	51100
STREAM AT PAR SANDS BEACH	81610603	208700	53200
CAERHAYS STREAM AT PORTHLUNEY BEACH	81811103	197484	41314
WEST WILDER BROOK AT ILFRACOMBE BEACH	73110303	251889	147831
COMBE MARTIN STREAM PRIOR TO BEACH	73110803	257459	147147
PILL RIVER AT BLUE ANCHOR	60580103	302720	143489
RIVER BRUE AT TIDAL SLUICE (BURNHAM JETTY)	60030125	331364	147252

The beach profiles, detailed maps of the sampling points and photos of the beaches are provided in Appendix A. The bacteriological and environmental data for each bathing water for 2000[34] to 2012 have been compiled and analysed by a database tool internal to the Environment Agency and provided in MS Excel ® spreadsheet format to the University of Exeter.

5.2.2 Sample observation dataset

To meet the BWD (European Commission, 1976) and rBWD (European Commission, 2006a), compliance sampling of bacteria counts from designated bathing waters is required on at least a weekly basis during the 20-week bathing season (15 May – 30 Sept). This started between 2000 and 2005 for the case study beaches. In 2012 the EA-E also began gathering daily samples (Monday – Friday) at five of the beaches, indicated in column 2 of Table 5.1. This provides useful additional data for the purpose of model testing.

Sample data (times of sampling, counts of bacteria from lab cultures from bathing water and salinity readings) are combined with meteorological data from nearby Met Office weather stations and tidal data (UK Hydrographic Office, 2014). These provide the time-variant dataset used for the machine learning models.

[34] *The column "Start for data model" in Table 5.1 indicates the year sampling commenced at each bathing water.*

The time-variant input and target features available as candidates for both the DT and ANN models are detailed in Table 5.3 and described in more detail in the following text:

Table 5.3. Description of Case Study Dataset Features

Feature ID	Feature Type	Sample Location	Feature Description	Feature Units
Timestamp	Input	Beach/River	Date and time of sample	millidays
TimeWRTHW	Input	Beach	Time of sample with respect to high water	hours
HtAtHW	Input	Beach	Height of water above mean sea level at high water	m
HtAtSample	Input	Beach	Height of water above mean sea level at time of sample	m
TidalRgAtSP	Input	Beach	Tidal range at standard port	m
TideLevelClass	Input	Beach	Tide classification [Spring | Mean | Neap]	m
AR24	Input	Raingauge	Antecedent rainfall total for 24 hours before sample	mm
AR48	Input	Raingauge	Antecedent rainfall total for 48 hours before sample	mm
AR72	Input	Raingauge	Antecedent rainfall total for 72 hours before sample	mm
AR96	Input	Raingauge	Antecedent rainfall total for 96 hours before sample	mm
AR120	Input	Raingauge	Antecedent rainfall total for 120 hours before sample	mm
Salinity	Input	Beach	Salinity (salt content of sample)	g l-1
	Target	Beach	Faecal Coliforms (FC) count	no/100ml
	Target	Beach	Faecal Streptococci (FS) count	no/100ml
NormalisedLogFC	Target	Beach	Normalised $\log_{10}$ of FC count	#
NormalisedLogFS	Target	Beach	Normalised $\log_{10}$ of FS count	#
FCpass	Target	Beach	FC count pass/fail	class
FSpass	Target	Beach	FS count pass/fail	class
BothPass	Target	Beach	Compliance pass/fail based on (FCpass AND FSpass)	class
Salinity	Input	River	Salinity (salt content of sample)	g l-1
	Target	River	Faecal Coliforms (FC) count	no/100ml
	Target	River	Faecal Streptococci (FS) count	no/100ml
NormalisedLogFC	Target	River	Normalised $\log_{10}$ of FC count	#
NormalisedLogFS	Target	River	Normalised $\log_{10}$ of FS count	#
FCpass	Target	River	FC count pass/fail	class
FSpass	Target	River	FS count pass/fail	class
BothPass	Target	River	Compliance pass/fail based on (FCpass AND FSpass)	class

Additional time-invariant data are also available describing characteristics of each stream catchment adjacent to each beach. This information could potentially be used in experiments to establish the effectiveness (or otherwise) of combined models that could make predictions for several beaches following a single calibration. This is beyond the scope of this thesis and is reserved for future research.

All data are normalised prior to use in the ANN models. Normalisation is either in the range [0...1] or [-1...+1] depending on feature. However Table 5.3 shows the measurement units in which each of the raw data features is recorded. Where features are qualitative, the coding scheme adopted is described below.

Timestamp

This is the date and time at which the sample has been collected. It is separately recorded for the beach and river samples. Recorded to 5-minute resolution, other possible sources of error include logging of an incorrect sample time or use of “standard” times of collection for each location. For ANN input, date and time is combined into a single real timestamp, then normalised between [0...1] for the set of samples used in the experiment.

Tide-related input features

TimeWRTHW: Time of sample with respect to high water is the offset in hours of the time at which the sample is taken with respect to the time of high water; negative values indicate sample collected before high water and positive values after high water. The potential range of values is therefore approximately [-6.00...+6.00]. Possible sources of error derive from incorrect recording of the time of sample and differences between the time of high water in the tide tables used and the actual time of high water at the beach. ANN input is normalised to the range of [-1...+1] over the set of samples used in the experiment.

HtAtHW: Height of water above mean sea level at high water is a measure of the tidal range at the beach, which can be a factor affecting tidal currents and so influence transport of bacteria to/from the bathing water. Height is measured in metres and is always a positive value above mean sea level. Possible sources of error derive from differences between actual height at the beach and at the nearest location for which tide tables exist. These can also be influenced by meteorological conditions. The ANN input is normalised between [0...1] for the set of samples used in the experiment.

HtAtSample: Height of water above mean sea level at time of sample is an interpolated value (in metres) from the tide tables based on the time of sample

with respect to high water (*TimeWRTHW*) and the height of water (in metres) above mean sea level at high water (*HtAtHW*). Potential source of error is compounded from all the above sources of error. ANN input range is normalised between [0...1] for the set of samples used in the experiment.

TidalRgAtSP: Tidal range at standard port is the range (in metres) of water level between high and low tide on the day of sample at the nearest standard port[35]. In south west England these include Dartmouth, Falmouth, Plymouth and Torquay (Admiralty, 2014). Values are taken from the Admiralty tide tables. Sources of error potentially include meteorological conditions. The ANN input is normalised between [0...1] for the set of samples used in the experiment.

TideLevelClass: Tide classification [spring | mean | neap] is a coarser representation of *TidalRgAtSP,* since it classifies the tidal range for the nearest standard port (and therefore the beach) into one of three classes: Spring (greater range than mean); Mean (approximately average range) and Neap (less range than mean). This therefore suffers from the aliasing error typical of any discretisation. For ANN input, the feature is coded [-1|0|+1] corresponding to neap, mean and spring tides.

Meteorological input features

AR24, AR48, AR72, AR96 and AR120: Antecedent rainfall total for 24,48,72,96 or 120 hours before sample are cumulative rainfall totals over the previous 1,2,3,4 or 5 days respectively, measured in mm. The rainfall readings are taken from the nearest available MetOffice raingauge. Sources of error include those well documented for raingauges (Habib et al., 2001) as well as those due to spatial variability in rainfall between the raingauge location and the stream catchment adjacent to the beach. The five ANN inputs are normalised between [0...1] for the set of samples of each given feature used in the experiment.

Other meteorological data could arguably be used to good effect, such as air temperature and pressure, windspeed and direction and wave height

[35] *a port whose tidal predictions are directly given in the Admiralty tide tables*

estimates, especially as these could potentially be readily available remotely, but these features were not supplied in the EA-E datasets, so are not included in this case study. Rainfall predictions could also be used to enhance the accuracy of daily bathing water quality predictions available to the public at designated beaches, although this is not attempted, because it would not alter the design of the ANN model substantially.

Salinity

The salt content (in mg/litre, equivalent to parts per thousand (ppt)) of the bathing water sample taken at the beach is recorded. Salinity is known to be a bacterial pathogen so may provide significant information as an input to the model. Where salinity is recorded indirectly as a conversion from measured conductivity, sources of error could arise from the calibration and measurement errors associated with the instrument. Where measured in the lab, any evaporation having taken place between collection and analysis is also a potential source of error. Typical range of salinity of sea water is 20 – 36 ppt. The ANN input is normalised between [0...1] for the set of samples used in the experiment.

Salinity is also recorded separately for the stream sample and would normally be expected to be a significantly lower value unless the river itself is tidal (e.g. East Looe, Mothecombe). The ANN input is normalised between [0...1] for the set of samples used in the experiment.

Target / output features

The target variable is the classification ('pass' or 'fail')[36] of a bacteriological bathing water sample, cultured in the laboratory and assessed against a compliance threshold of 500 *Faecal Coliforms*/100ml (FC) and/or 200 *Faecal Streptococci*/100ml[37] (FS).

The bacterial counts for (FC/EC) and (FS/IE) vary over a range of several orders of magnitude. A lower limit of 10 is set and anything less than this is

[36] *The nomenclature used throughout uses "pass" as equivalent to positive and "fail" as equivalent to negative. This is in order to maintain consistency with the work done at the EA-E on the DT models.*

[37] *A count above these thresholds of either or both of these organisms is treated as a "fail"*

recorded as “<10”. Where regression models are constructed to predict the bacteria counts directly, counts are first converted to $\log_{10}$ values and then normalised in the range [0..1] over the set of samples used in the trial. These features are designated *NormalisedLogFC* and *NormalisedLogFS*.

For classifier models, the target values are coded using 0=”fail” and 1=”pass” by comparing the individual bacteria counts with the compliance threshold to produce (*FCpass*, *FSpass*) target classification labels for each individual bacteria species then for the single target feature BothPass: = (*FCpass* AND *FSpass*). Potential sources of error arise from the laboratory procedures for culturing and assaying the samples (Watson et al., 1977) as well as with the procedures for collecting the seawater and river water samples and their transport back to the lab. For the ANN, the [0|1] values are used as targets both for training and evaluating test performance. For the ANN output the classification decision threshold is varied using a Receiver Operating Characteristic (ROC) approach described in section 5.2.5.3.

Separate models would be constructed to predict stream sample compliance and beach sample compliance. However, they would use the same feature designations for convenience.

Sample selection strategy

Samples exceeding the bacteriological threshold in dry weather (quantified by a 96hr rainfall total <5mm) are removed from the dataset, since these would have been caused by events independent of the rainfall and tidal data, e.g. wrongly connected waste water systems or bird or dog fouling in the catchment. Additional predictor input features would be needed if these were to be included in the models.

5.2.3 Decision Tree models

The models described in this section are the subject of previous studies conducted by the Environment Agency between 2007 and 2009 (Tyrrell, 2010), so should not be taken as a contribution in this thesis. They are included for the purposes of comparison. Due to the statutory requirement in the rBWD to provide public advisory warnings in the event of bacteria counts exceeding

statutory levels; this is treated as a classification problem. Predictive models for eight bathing waters were built and tested in 2012. The Environment Agency's DT models are built using a script developed from the Classification Trees module within IBM SPSS™ Statistics software (IBM, 2011; Mola, 1998). The procedure creates a tree-based classifier by taking a set of data points and grouping them into categories of a dependent (target) variable based on values of a number of independent (predictor) variables, which form the inputs to the model. The predictor variables used in the original EA-E study are: *antecedent rainfall totals for 24hrs, 48hrs, 72hrs, 96hrs, and 120hrs, tidal range,* and *tidal state*. Compliance samples have been gathered weekly during the 20-week bathing season since 2002 and this data is also available for the ANN experiments.

Of the various tree growing methods available in SPSS™, previous studies have shown that the CART (Classification and Regression Trees) method gives the most accurate results (McPhail and Stidson, 2009). CART is a non-parametric algorithm that produces a binary decision tree constructed by splitting each node into two child nodes repeatedly, beginning with the root node (parent) that contains the whole training data set The data are split into segments, each of which is as homogeneous as possible with respect to the target variable. Each branch of the tree ends with a terminal node which is uniquely defined by a set of rules that may then be applied to predict future events. The complexity of the tree depends on the underlying distribution of data, and it follows that the stronger the relationship between dependent (target) and independent (predictor) variables the simpler the finally constructed tree is likely to be. By building trees automatically of sufficient complexity, it is possible to classify virtually every sample in the training dataset correctly[38]. However, such DTs tend to suffer from a problem analogous to overfitting in ANNs in that they do not generalise their classifications well for new test data, not included in the original training set In order to alleviate this issue, DTs are normally pruned so as to achieve acceptable validation results, even though achieving an above-zero level of misclassification on the training set. An example decision tree is presented in Figure 5.3.

[38] *An exception to this is where two samples have the same input feature values, but different target values. These would not be able to be separated by the DT model.*

The original EA-E study focuses on correctly predicting poor water quality and protecting public health, but also on satisfying the needs of beach managers to minimise the number of public advisories per bathing season. Therefore the models are weighted to minimise the number of incorrectly predicted exceedances of the bacteriological threshold (false negatives).

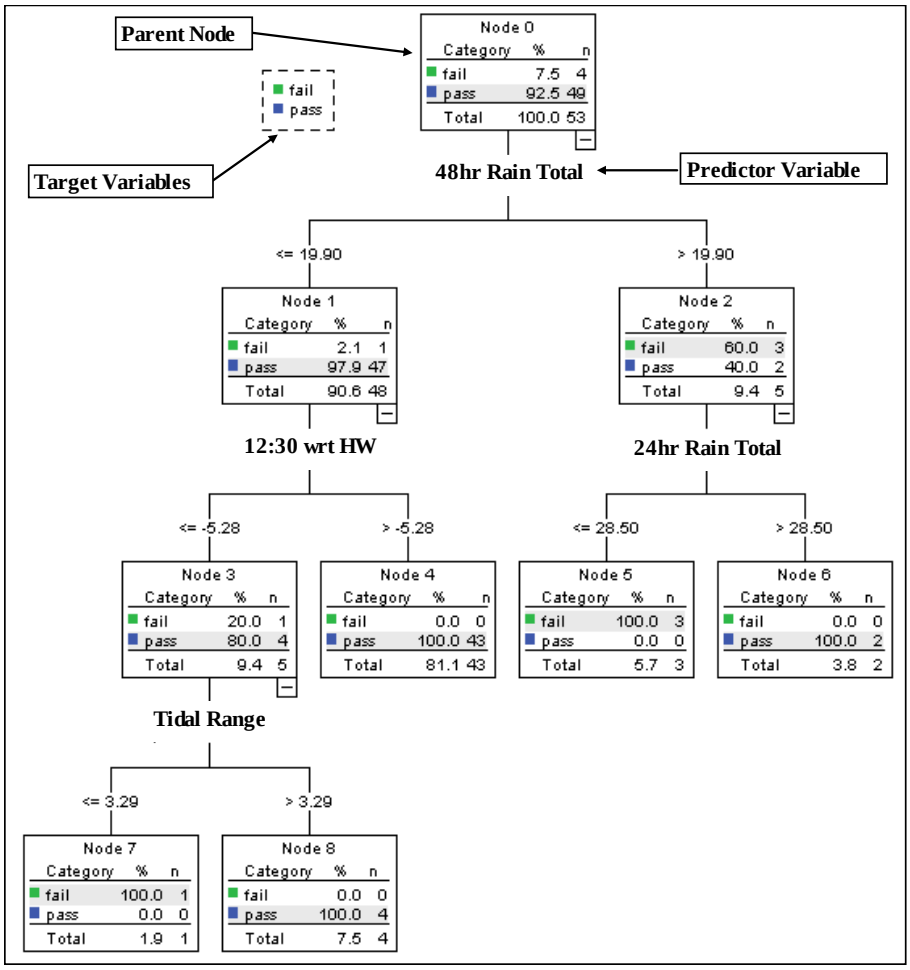


Figure 5.3. Example Decision Tree

In the Environment Agency experiments, an initial decision tree is built for each bathing water using the data from 2000 to 2006, and the resulting rules from the terminal nodes are applied blindly to the 2007 data for validation. The models are then rebuilt including the 2007 data and resulting rules applied blindly to the 2008 data. This iterative process of calibration and validation is continued until all the data to 2011 is included in the model, giving a total of five validation trees per bathing water plus a tree for investigational use in the 2012 bathing season. The Environment Agency's DT model results for the five

bathing waters[39] for which daily sampling data are available for the 2012 bathing season are presented for comparison to the ANN results in section 5.3.

5.2.4 Simple trigger models

Research at the Environment Agency, together with the immediate requirement to provide live early warning systems at the designated bathing beaches to meet the requirements of the rBWD, motivated an investigation into whether so-called "simple trigger" models could be effective. Simple trigger models can be regarded as DT models pruned to a single node. These are based on generating warnings if a given threshold of antecedent rainfall occurs. The threshold(s) and the antecedent periods required are found to be dependent on the properties of the catchment at each beach. Typically a 10mm threshold of 24-hour antecedent rainfall is found to perform well. However, thresholds of 8 and 15mm are also used, as are 24 and 48-hour antecedent periods. These are documented in the summary table of results for all models and all 5-beaches. They are included here for comparison with the DT and ANN models.

5.2.5 ANN models

5.2.5.1 Aim of case study ANN trials

The aim of this case study is:

5. To evaluate performance of NFCV ensembles of ANNs used as binary classifiers to predict bathing water quality exceedances at five beaches in south west England using an ROC scenario.
6. To evaluate effectiveness of NPSFS methodology to select relevant input features to the ensembles and produce a reduct[40] input feature set for which NFCV ensemble performance is as good or better than the original with the full input feature set.

[39] *These are indicated in Table 5.1.*

[40] *i.e. a reduced input feature set based on a feature selection strategy.*

7. To compare performance of DT, Simple Trigger and ANN models at the 5-beaches using the 2012 bathing season daily sample dataset.

5.2.5.2 Description of NFCV / NPSFS approach

Using the same datasets, separate ANN classifier models are built for each beach and tested using MATLAB ® V2012a (Mathworks, 2012). The models are based on the RAPIDS package developed by the author and described in earlier publications: (Duncan et al., 2011, 2013a, 2013c). The same automated Neural Pathway Strength Feature Selection (NPSFS) methodology as described in Chapter 4 is implemented. This uses ANN ensembles with N-fold cross-validation (NFCV) in which each bathing season (year) 2000-2011 of observations is treated as a separate data fold. Thus, for each beach, an ensemble of 12 ANN models is trained and then each is tested on a different remaining ("left-out") fold (bathing season of samples). Finally the bathing season for 2012, consisting of 100 daily samples (collected Mon-Fri each week for the 20 weeks of the bathing season) is used as the test fold for the entire ensemble of ANNs.

In order to evaluate the optimum architecture for the ANNs, ensembles are constructed with a range of numbers of hidden units. For consistency, the results presented use the values of 5, 8, 12, 18, 27 and 40 for the majority of the experiments. Given the use of 12-input features and a single output node, application of Han's rule-of-thumb for optimal number of hidden units (Han et al., 2007; Han, 2003) would yield $N_{HU}=(12+1)x2/3 \approx 8$.

5.2.5.3 Receiver Operating Characteristic (ROC) approach

ROC is a standard approach to optimisation of binary classifiers, derived originally from the need to optimise radar radio receivers (from which it derives its name) during WWII, so as to minimise misclassifications (both false positives and false negatives) by varying the detection threshold. In a current machine-learning context, Fawcett (2006) provides an excellent review and introduction to ROC curves. The standard ROC approach applies to binary classifiers, but it can be extended to those with multiple classes. However, the number of dimensions rises as a function of $n\ (n-1)$, where n is the number of classes. A

short description of the ROC curve and its construction in the context of an ensemble of ANN binary classifiers follows:

Figure 5.4 shows a typical set of ROC curves for an ensemble of 12 ANN classifiers (ANN2000 – ANN2011). The example is taken from one of the beach case studies in this project. The axes of the ROC curve are False Positive Rate (FPR) x-axis and True Positive Rate (TPR) y-axis. These vary between [0...1] in both cases, so the area of the entire plot is exactly 1. TPR and FPR are defined as follows (Fawcett, 2006):

$$TPR = \frac{TP}{P} = \frac{TP}{(TP + FN)} \qquad (5.1)$$

$$FPR = \frac{FP}{N} = \frac{FP}{(FP + TN)} \qquad (5.2)$$

where: TP = the number of true positive samples (correctly predicted by the model); P is the number of positive samples, which includes true positives and false negatives (FN); FP = the number of false positive samples (incorrectly predicted by the model); N is the number of negative samples, which includes false positives and true negatives (TN).

For completeness, we also have: $TNR = (1 - FPR)$ and $FNR = (1 - TPR)$ for true negative rate (TNR) and false negative rate (FNR).

The ideal operating point for any model is the point in the top-left corner of the ROC plot; i.e. FPR=0 and TPR=1. However this is rarely achieved for real models and datasets with many samples in each of the 2 classes.

Although the activation function for the ANN output is chosen so that it tends to output values close to 1 or close to 0 for its predictions, the value actually output in response to each input sample is a real number between 0 and 1. The ROC curve is constructed by varying the decision threshold used to discriminate between the 2-class labels and applying it to the ANN's output signal. We use "pass" as positive and "fail" as negative throughout in order to be consistent with the earlier work on Bacti at the Environment Agency.

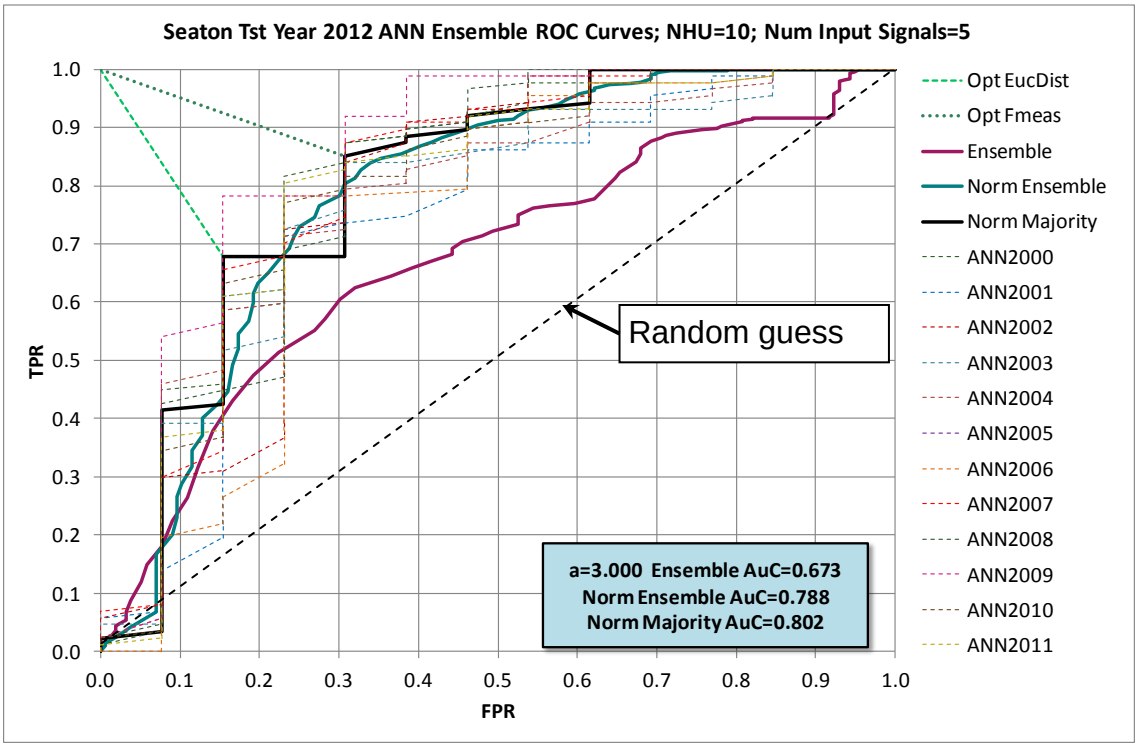


Figure 5.4. ROC Scenario for an Ensemble of ANN Classifiers

When the threshold is greater than the maximum ANN output value, all samples are classified as negative ("fail") regardless of the target classifications. Therefore the true positive rate is zero and the false positive rate is also zero ([0, 0]). Conversely, when the threshold is less than the minimum ANN output value; all samples are classified as positive ("pass") regardless of the target classifications. Therefore the true positive rate is one and the false positive rate is also one ([1, 1]). With threshold values in the range between these two, an arc is described as shown in Figure 5.4. This shows a total of 15 such ROC curves. Regardless of the relative sample counts for the two classes, a process of random guessing yields a straight diagonal line between the points [0, 0] and [1, 1]. Any classifier yielding an ROC curve above and to the left of this line is better than random guessing. The reason for the stepped appearance of the ROC curves illustrated is due to the small number (*N=13*) of negative samples in the ensemble test dataset (total 100 samples).

In order to illustrate this further, Figure 5.5 shows the raw data behind the construction of an ROC curve. The x-axis represents the index of the 100 daily samples from the 2012 bathing season in the ensemble test dataset being applied to the model. The y-axis represents values of threshold applied as the

decision point applied to the ANN output to distinguish between positives (pass) and negatives (fail). In the plot area, for each sample, false positives are displayed in green, false negatives in red and true positives and true negatives in grey.

For negative samples:

- There is a green bar at the bottom where the ROC is classifying the sample as positive, due to the threshold being below the actual output value of the ANN. This is a false positive
- There is a grey region above the green bar, where the threshold is above the ANN output for the sample and it is correctly classified as negative. This is a true negative.

For positive samples:

- There is a red bar at the top, where the threshold is above the actual output value of the ANN, so the sample is classified as negative. This is a false negative.
- There is a grey region below the red bar, where the threshold is below the ANN output for the sample, so it is correctly classified as positive. This is a true positive.

As can be seen, a trade-off exists between many false negatives at the top (threshold too high) and false positives at the bottom (threshold too low). For a perfect model, there would be no overlap between the red and green zones; in practice the optimum threshold corresponds with a balance of minimised false positives and negatives (here a value around 0.7 to 0.75).

Another feature of the dataset that can be observed from this chart is the skew between the 2 classes of samples, with far more positive (“pass”) samples (red/grey) than negative (“fail”) samples (grey/green).

For each value of threshold, from this data TPR and FPR can be calculated using equations (3.1) and 0 and the ROC curve plotted.

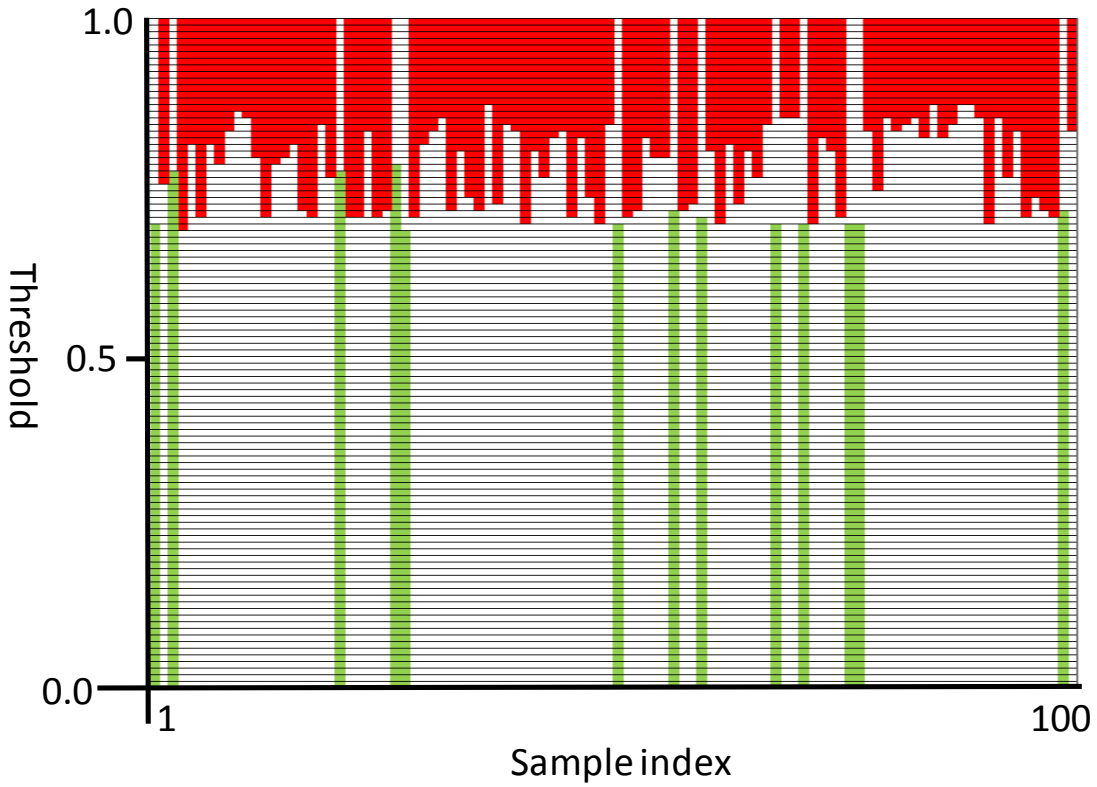


Figure 5.5. ROC false positives and false negatives by threshold and by sample

Area under the Curve (AuC) metric

A metric is required to evaluate the overall performance of a classifier or ensemble of classifiers. A commonly used metric for this purpose is the area under the ROC curve (AuC)(Bradley, 1997), which, for an ideal classifier would be 1.0 and for a random guess process would be 0.5. The advantages of AuC over single threshold measures of TPR and FPR are that it is threshold-independent and it allows the optimum operating point (decision threshold) to be discovered, since the shape of the whole curve is then known.

Optimum operating point location

This optimum point can be determined in a number of ways. Two methods are used here.

1) To evaluate the point with highest value of modified F-measure.
2) To evaluate the point on the ROC curve of minimum Euclidean distance from the ideal point *[0, 1]*.

1) Stidson et al. (2012), in their study relating to quality at Scottish bathing waters, propose use of a modified *F* measure to evaluate model performance

using a weighting (a=4 in (5.3)) to minimise the number of incorrectly predicted passes (levels below the bacteriological threshold): False Positives (*FP* in (5.3)). This effectively weights public health risks as 4 times more important than economic risks to tourism. The same value of '*a*' is used here.

$$F = \frac{(1+a)TN}{(1+a)TN + aFP + FN} \qquad (5.3)$$

where: *F* = modified F measure; *TN* = number of true negative samples; *FP* = number of false positive samples and *FN* = number of false negative samples (negative = fail; positive = pass). Using a finite step-size for threshold value, the F-measure can be calculated for all threshold values on the ROC curve and *locmax{F}* located, to identify the optimal value of threshold. From the ROC curve, the corresponding values of FPR and TPR can also be found.

2) As an alternative, the '*a*' weighting is adapted for use with ROC curves by effectively stretching the x-axis (*FPR*) for values of *a>1* and shrinking it for *a<1*. The Euclidean distance (*E*) of each point to the ideal *[FPR=0; TPR=1]*, on the scaled ROC curve is calculated using the scaled x-axis and the optimum operating point (E_{opt}) determined using (5.4).

$$E_{opt} = min\left\{\sqrt{(a \cdot FPR)^2 + (1-TPR)^2}\right\} \qquad (5.4)$$

where: E_{opt} = optimum Euclidean distance; *FPR* = false positive rate; *TPR* = true positive rate; *a* is the same weighting factor as in Stidson et al. (2012) and { } indicates the set of *E* values calculated for all threshold values on the ROC curve. *(1 – TPR)* = *FNR* as previously stated.

Sensitivity analysis for optimum operating point location

The 'a' weighting has an effect of the location on the ROC curve of the choice of optimum operating point. A sensitivity analysis is conducted by varying 'a' between 0.2 and 5.0 to observe the effect on location of optimum point. This is conducted both for use of the maximum of F-measure and the use of minimum of Euclidean distance to locate these points.

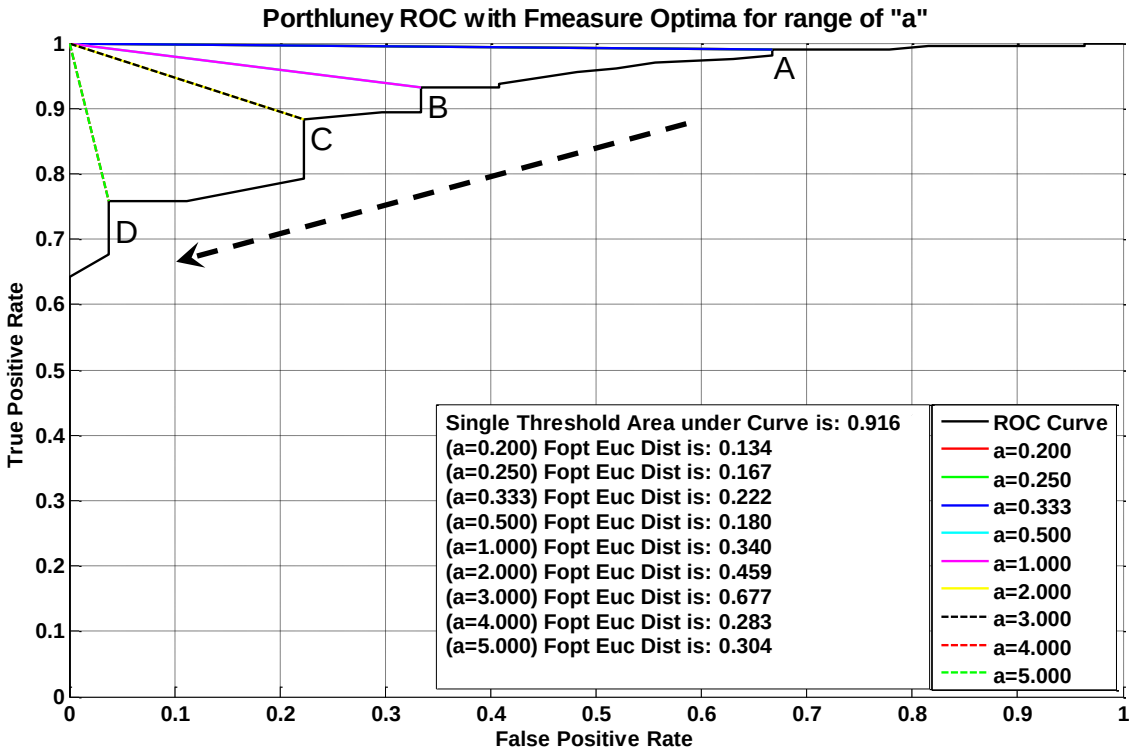


Figure 5.6. Sensitivity analysis: Effect of varying 'a' on optimum operating point, using F_{max} as criterion

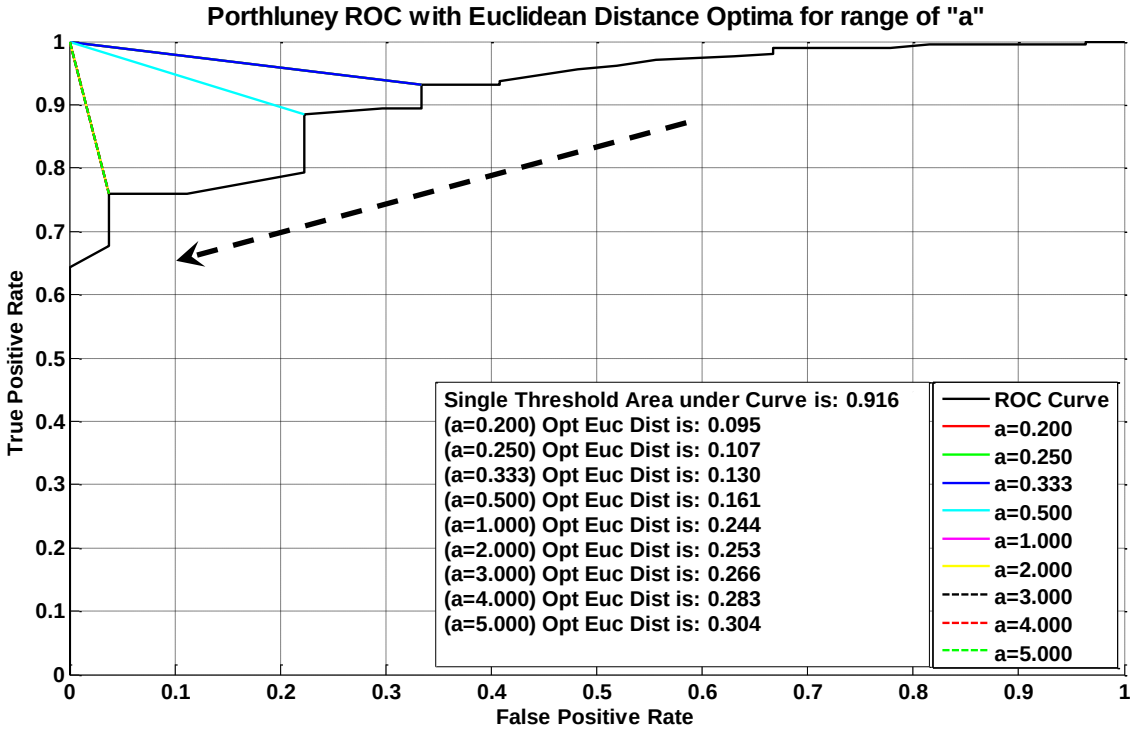


Figure 5.7. Sensitivity analysis: Effect of varying 'a' on optimum operating point, using E_{opt} as criterion

Porthluney beach is used as an illustrative example in this analysis. An ROC curve is constructed indicating FPR and TPR as threshold varies on the ANN output, in the way described above. 'a' is then varied between 0.2 and 5.0 and a line connecting the ideal point [0, 1] and the location of the optimum operating point is plotted for each. The Euclidean distance (scaled length of this line) is tabulated on the chart on each case. Figure 5.6 is constructed using the maximum value of the F-measure to locate the optimum point for each value of 'a'; whereas Figure 5.7 uses minimum Euclidean distance (E_{opt}) instead of the F-measure. As can be seen from the dashed arrows, when 'a' is small, the optimum operating point is far to the right, with very high values of TPR and consequently also FPR. As 'a' increases, the optima move downwards and to the left and TPR and FPR both decrease. However, due to the shape of the ROC with convex corners (A, B, C, D in Figure 5.6) the progress of optimum is not smooth; rather, the optimum switches from corner to corner on the ROC at certain critical values of 'a'. As a result, some of the lines shown in the key are obscured behind the other lines. The colour of the line showing connected to each corner indicates the highest value of 'a' to locate the optimum at that corner as 'a' increases, before it switches to the next corner. From this analysis, it can be seen that, for the same value of 'a', the maximum F-measure criterion consistently chooses points further upwards and to the right than when minimum Euclidean distance is used.

5.2.5.4 ANN methodologies used for Bacti case studies

The methodologies applied follow that described in Chapter 4 with the addition of the ROC analysis of ANN binary classifier outputs[41]. In order to achieve this, two alternative training algorithms (5.8 and 3.4) are compared.

Two-layer fully-connected feedforward ANNs are used. Inputs are not time-lagged explicitly, due to the very long timestep for the samples[42]. Instead, the implicit time-lagging present in the antecedent rainfall totals is exploited. Figure 5.8 and Figure 5.9 illustrate two alternative architectures for the learning schemes used for training the ANNs. Use of AuC or FPR and FNR of an ROC

[41] The ANNs used in the case study of chapter 4 are regression models, used to predict a continuous quantity: log (fire area+1).

[42] Samples are daily or weekly. The number of timesteps in the input time-window is therefore 1. This ensures independence of the samples from each other without the need to parallelise samples in a time window.

curve are not standard network performance functions in the MATLAB NN-toolbox, so the alternative training methodologies replace the MATLAB standard training algorithms.

In both cases, N-fold cross-validation (NFCV) is used as in Chapter 4. An ensemble of 12 ANNs is thus constructed. This number is determined by the number of years of bacterial sampling data available from the Environment Agency. Each year (bathing season) is used as a separate data fold. In the initial trials, for each member of the ensemble, the same set of (coincidentally) 12 input features[43] is applied to the ANN; these are listed in the top section of "Table 5.3. Description of Case Study Dataset Features" and further information is provided in section 5.2.2.

Also in both cases, the number of neurons in the hidden layer is varied during experimentation in order to determine optimum network architecture. For each trialled number of hidden neurons a full NFCV ensemble of ANNs is constructed. For each ANN, the real-valued classification responses from the single output neuron are compared to a (ramped) set of threshold values covering the span of [0...1] in 101 steps, and the area under the ROC curve (*AuC*), *F* (equation (5.3)) and E_{opt} (equation (5.4)) are computed as performance metrics. For more information on ROC curves, details are provided in section 5.2.5.3.

Early Stopping

For both the NSGA-II and SCG algorithms, early stopping is employed in accordance with best practice. Validation is performed by interrupting the training process regularly after every given (configurable) number of epochs. During the interruptions, progress of training is validated using a "validation" fold of the data excluded from both the training and the test datasets for the current ANN ensemble member. For NSGA-II, the ‘best’ solution is selected from the population by using minimum of Euclidean Norm of training error in the 2-objectives. This is then used for validation. For SCG, there is only a single ANN solution to validate. Thus the metric used for the validation is algorithm-

[43] The datasets for Par and Readymoney have 8-input features.

dependent. At each validation step, the validated ANN network is saved if its performance is better than the previously recorded lowest validation error.

The conditions for stopping the training process are as follows:

a. if the running-averaged validation error increases by a factor of greater than 1% over the minimum error so far achieved; or
b. if the (configurable) maximum number of training epochs is reached; or
c. if the (configurable) goal of error performance is reached; or
d. if validation error has stagnated at a fixed level for more than 4 validations.

When stopping occurs, the current or previously recorded network with the lowest validation error is taken as the trained ANN for inclusion in the ensemble.

Dual Objective Evolutionary Algorithm Approach to Training

The methodology described here relates to earlier work conducted by Anastasio and Kupinski (1998) in which they use a multi-objective genetic algorithm (MOGA) to optimise rule-based and hybrid rule-based / ANN classifiers to detect presence of cancerous lesions in mammary and chest tissue. Their methodology uses free-response ROC curves (FROC) (Chakraborty, 1989; Penedo et al., 2005), which allow the detection of multiple regions of diseased tissue simultaneously. However, it is equally applicable to the standard ROC curves used here. They demonstrate that their multi-objective approach is superior to results produced by varying each of their models' input features individually. Their work pre-dates the creation of NSGA-II (see below) but nonetheless the GA they use implements a similar dominance-based ranking scheme, binary tournament selection and crowding distance (which they call niche radius)).

Further examples of the use of ROC curves together with multi-objective evolutionary algorithms (MOEAs) are contained in Reckhouse (2010) PhD thesis involving optimisation of Short Term Conflict Alert (STCA) air traffic safety related systems to locate the optimum operating point. This simultaneously

minimises nuisance alerts and maximises genuine alerts. Additionally the shape of the whole ROC is characterised by applying offline perturbations of classifier parameters not permitted in the online STCA system. Similarly, Evolutionary Strategies (a variant of MOEAs) are applied to the STCA optimisation problem using ROCs (Everson and Fieldsend, 2006; Fieldsend and Everson, 2002).

Figure 5.8 illustrates the schematic for the Evolutionary Algorithm-implemented ANN training methodology. This approach is applied to optimisation of the ANN weights and biases and uses a dual-objective realisation of the popular evolutionary algorithm NSGA-II (Deb et al., 2002b) in which decision variables are real-valued and represent ANN weights and biases. NSGA-II can now be regarded as an “industry standard” and is described in the literature review of Chapter 2. This algorithm completely replaces the training functions provided as standard in the MATLAB NN-toolbox. Algorithm 5.7 defines the process.

Crossover is accomplished by swapping real values (ANN weight and bias values) between chromosomes of the parent solutions. Two types of mutation are tried: replacement mutation and Gaussian mutation. In replacement mutation, decision variables selected for mutation are simply replaced with new real values selected at random from a uniform distribution. In Gaussian mutation, a delta quantity selected from a zero-mean Gaussian distribution is added to the existing decision variable value.

Algorithm 5.7: NSGA-II: non-dominated sorting genetic algorithm II (Deb et al., 2002b) – adapted for Bacti case study

Input: Bacti dataset for a given beach (sections 5.2.1 & 5.2.2); initial populations P_1 and Q_1 of N/2 real-valued chromosomes each (decision variables → ANN weights and biases)

Output: Optimised Population P_g with ranks, crowding distances, domination lists and fitnesses for each member.

1. g ←1 first generation / epoch
2. **Begin**
3. $R_g \leftarrow P_g \cup Q_g$ combine parent and offspring population
4. Evaluate fitness of all members of R_g
5. fast-non-dominated-sort (all non-dominated fronts F_i of R_g)
6. Initialise P_{g+1} ← [ ]; *i=1*
7. **Begin**
8. Add members of each front F_i until the P_{g+1} is (almost) filled
9. Calculate crowding-distance in F_i
10. $i \leftarrow i + 1$ check the next front for inclusion
11. **End;**
12. Sort F_i in descending order using partial order (rank, crowding distance)
13. choose the first (N - $|P_{g+1}|$) elements of F_i to fill parent population P_{g+1}
14. Use selection, crossover and mutation to create new population Q_{g+1}
15. g ← g+1 increment the generation counter
16. **End;**

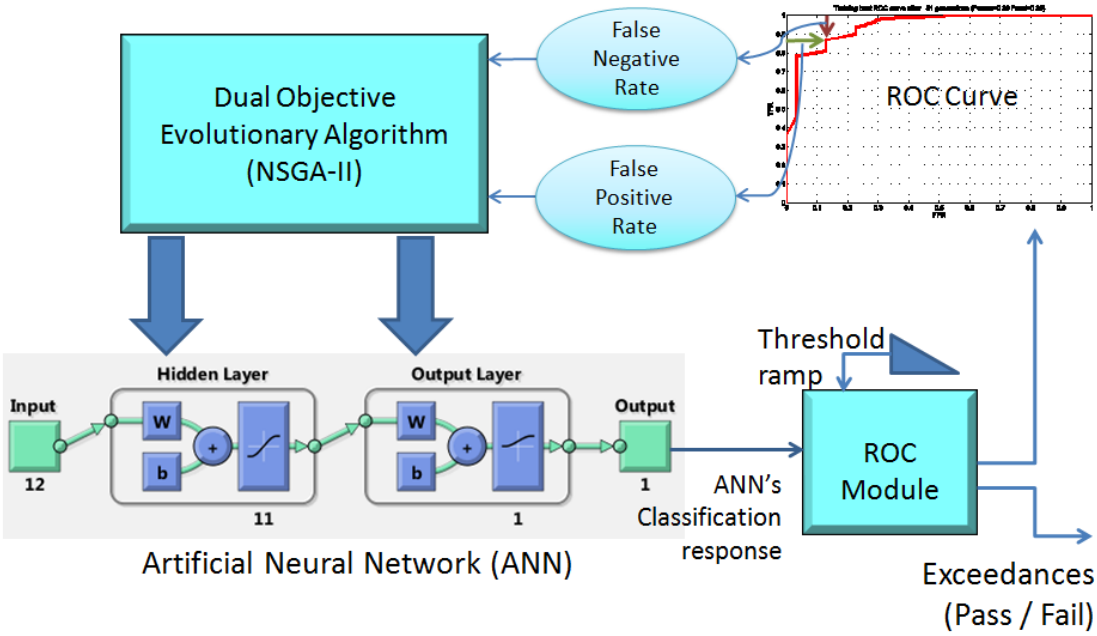


Figure 5.8. EA-based ANN Training Architecture with ROC Scenario

The two objectives evaluated by the objective function are based on minimization of cost in both cases: False positive rate (*FPR)* and false negative rate (*FNR)*. These are derived by constructing the ROC and using equation (5.4) with a=1 to discover the optimum operating point; hence yielding values

for FNR and FPR. These two objectives are truly traded off against each other as illustrated by moving along the x-axis of Figure 5.5. Moreover, the optimisation of the both FNR and FPR together should lead to further separation between the false positive and false negative classes illustrated in Figure 5.5.

In the decision space, each population member's chromosome consists of values ($[-W_{span}...+W_{span}] \in \mathbb{R}^N$), where N is the total number of ANN weights and biases. This can be viewed as an N-dimensional decision space for each member of the population. If ANN architectures are allowed to vary (e.g. number of hidden units) within the population, this means that the number of decision variables may vary from member-to-member. This leads to a challenge with regard to reproduction, since chromosome lengths may be different, making crossover difficult. Solutions for this exist within the literature. However, this problem is avoided here by ensuring that all candidate ANN architectures are the same within any one training run and hence its population of ANNs. W_{span} is a constant set limit for the value range of each weight or bias for a given experiment. Values of W_{span} typically vary in the range [1, 10].

The use of evolutionary algorithms (EAs) for the training of ANNs is well discussed in the literature and reviewed in Yao (1999). This is also covered in section 2.4 of the literature review in Chapter 2. NSGA-II uses a binary tournament selection. At each generation (equivalent to an epoch of the training), fitter members of the population have a higher probability of being selected as parents for reproduction of the new child solutions for the next generation. It is also classed as a domination-based algorithm; that is that it maintains ranks of domination for each population member. Rank 1 is non-dominated; rank 2 is dominated by one other solution etc. In addition to performance in the two objectives, individuals with larger crowding distances between them and their nearest neighbours on the same rank have a higher probability of selection for reproduction. As a result, the selection pressure is maintained and generations progress in the direction of improved fitness (lower costs) in both of the two objectives described above as well as in terms of maintaining a distribution of solutions along the Pareto front.

A population size of 100 is found to be adequate and probabilities of crossover and mutation are also varied during experimentation to determine reasonable value ranges. These are both found to affect rates of convergence.

Decision Making

Upon completion of EA-based training, a "best" solution needs to be selected from the population of ANN solutions for use as the NFCV ensemble member. In the literature, the conventional approach is for a human Decision Maker (DM) to make this selection based on the set of non-pareto-dominated solutions from the population (Bechikh et al., 2011; Das, 1999; Ferreira et al., 2007; Rachmawati and Srinivasan, 2009). However, in order to automate the algorithm fully, this is selected using the criterion of the (non-pareto-dominated) ANN with the minimum Euclidean Norm of cost with respect to both *FPR* and *FNR* i.e. the two objectives.

Other strategies to select a "best" solution could also be adopted, such as looking for the "knee" in the Pareto front of non-dominated solutions (Branke et al., 2004; Rachmawati and Srinivasan, 2009).

Validation during training and early-stopping are also used in line with best practice. When using the EA, validation is performed periodically after a number (V) of epochs / generations of the population[44] using the same Euclidean Norm criterion to select from the population the "best" ANN to evaluate.

EA-based Bacti NFCV ensemble Algorithm

Algorithm 5.8 defines the methodology used for the EA-based section of the trial. This brings together the methods and techniques described in the earlier subsections of this section. Results are presented in section 5.3.2.

[44] *This is generally kept at 10 generations, as a trade-off between execution speed and stopping as soon as possible after the optimum point has been reached.*

Algorithm 5.8: EA-based NFCV ensemble EQR input feature selection trial

Input: Bacti dataset for a given beach (sections 5.2.1 & 5.2.2); configuration file (Chapter 4 "*ANN architecture and configuration setup*")

Output: set of evaluations of feature selection methodology and a reduced input feature set ensemble of ANN classifier models; performance evaluation for ensemble using ROC *AuC,* F_{opt} and E_{opt} metrics.

1. *For each ANN architecture (number of hidden units):*
2. **Begin**
3. Create a NFCV ensemble of 12 members using the strategy described in Chapter 4 and using the first 12 of the 13 data folds described above.
4. *For each ensemble member:*
5. **Begin**
6. *Use NSGA-II algorithm 5.7 to train and select ANN ensemble member as follows:*
7. **Begin**
8. Create parent population of *N=100* weight/bias vectors and randomly initialise
9. Instantiate ANN object with appropriate architecture
10. *For each parent population member, evaluate fitness of ANN*
11. **Begin**
12. Set ANN object weights and biases ← weight / bias vector
13. Simulate ← training dataset
14. Construct ROC and locate optimum Euclidean distance (equation 0)
15. Return FNR and FPR for optimum Euclidean distance
16. **End;**
17. Train for up to G==100 generations/epochs (g) using batch-mode offline training (Algorithm 5.7)
18. *Interrupt for validation every V generations*
19. **Begin**
20. Select "best" population member to use for validation (equation 0)
21. Simulate ← validation dataset for this ensemble member / data fold
22. Construct ROC and locate optimum Euclidean distance (equation 0)
23. IF (early-stopping criterion met OR G==100) record "best" ANN and exit NSGA-II
24. **End;**
25. **End;**
26. On completion of training, simulate with the trained network using the 13th (2012) ensemble evaluation data-fold and store responses together with ROC evaluation metrics
27. Store the trained weights and biases and combined pathway strength matrix
28. **End;**
29. Evaluate overall ROC performance of ensemble using collation of ANN results
30. Evaluate EQR for each input feature; analyse pathway strength vectors over the ensemble and rank the inputs in descending order of EQR
31. **End;**
32. Assess mean and median rank for each input over all ensembles / ANN architectures
33. Repeat once from 2. using reduct of only input features with mean EQR>0
34. Repeat once from 2. using reduct of only the 6 highest mean ranked input features
35. Compare ROC *AuC,* F_{opt} and E_{opt} metrics for the full 12 input features trial with those for the reduct trials using Student's T-test (Fay and Proschan, 2010)

SCG-based Approach to Training

As the second ANN training alternative, a Scaled Conjugate Gradients (SCG)(Møller, 1993) based method for optimisation of ANN weights and biases during training is also employed. This is illustrated in Figure 5.9. The SCG algorithm is discussed in section 2.2.5 of the literature review.

Conventionally, a metric such as Mean Squared Error (MSE) or Mean Absolute Error (MAE) between the ANN outputs and the target values over the training dataset is used in the objective function. These performance functions are a standard feature of the MATLAB NN-toolbox.

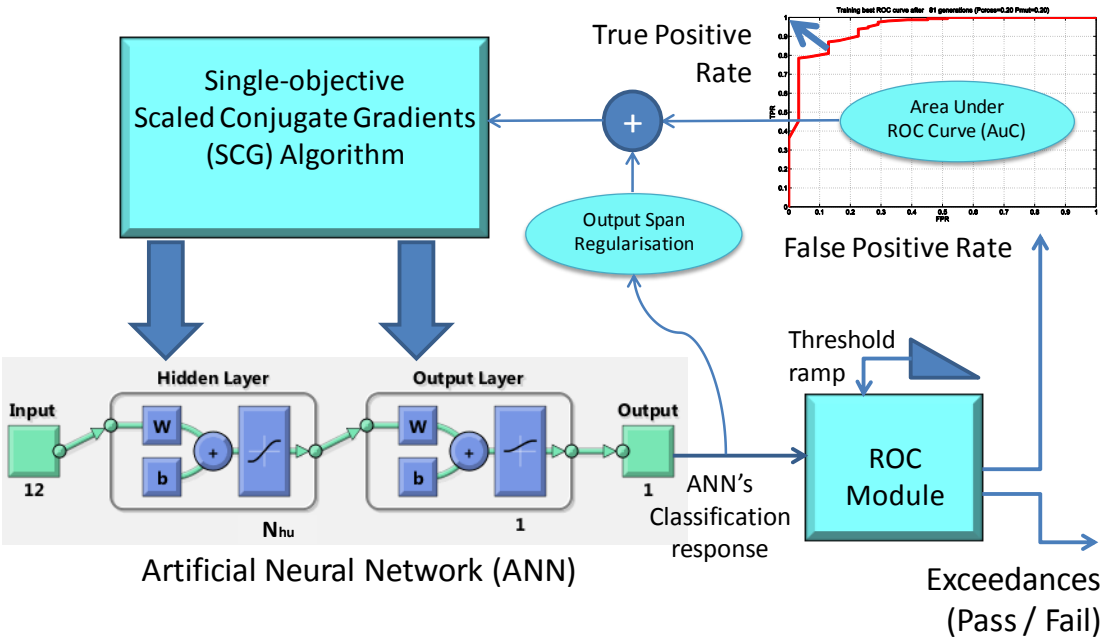


Figure 5.9. SCG-based ANN Training Architecture with ROC Scenario

Instead, here, a user-defined function is overloaded in place of the MSE function. The user-defined function constructs an ROC curve and computes the area under the curve *(AuC)*. The measure *(1 – AuC)* is used as a cost for minimization[45]. The *AuC* is calculated using the Trapezoidal Rule with 100 intervals. The gradient of the objective function *(1 – AuC(W))* with respect to *W* is calculated using a finite difference approximation.

[45] *This is effectively the area above and to the left of the ROC curve.*

Additionally, for the benefit of constructing a majority decision ROC for the entire ensemble of ANNs it is desirable to encourage all ANN output spans to approach a normalised range of [0...1]. Therefore an additional objective term is created: a measure of (1 – Span of ANN output) is combined (using equal weighting) with (1 – AuC) to produce an overall cost function. This is a single-objective approach as these two costs do not exist in a trade-off with each other, but are mutually reinforcing. This is indicated in Figure 5.9 by the summation term (+) in the feedback loop returning as input to the optimiser.

SCG-based Bacti NFCV ensemble Algorithm

Algorithm 5.9 defines the methodology used for the SCG-based section of the trial. The performance functions used in the two training approaches are deliberately chosen to be different, in order to demonstrate both using ROC AuC in a single objective optimisation and using false positive rate and false negative rate in a dual-objective scenario. Additionally, comparisons between evolutionary algorithm and SCG approaches are able to be made.

Results are presented in section 5.3.2.

Algorithm 5.9: SCG-based NFCV ensemble EQR input feature selection

Input: Bacti dataset for a given beach (sections 5.2.1 & 5.2.2); configuration file (Chapter 4 "*ANN architecture and configuration setup*")

Output: set of evaluations of feature selection methodology and a reducted input feature set ensemble of ANN classifier models; performance evaluation for ensemble using ROC *AuC,* F_{opt} and E_{opt} metrics.

1. *For each ANN architecture (number of hidden units):*
2. **Begin**
3. Create a NFCV ensemble of 12 members using the strategy described in Chapter 4 and using the first 12 of the 13 data folds described above.
4. *For each ensemble member:*
5. **Begin**
6. *Use SCG algorithm to train each ANN ensemble member as follows:*
7. **Begin**
8. Instantiate ANN object with appropriate architecture and uniform randomly initialise weights and biases
9. *Evaluate fitness of ANN*
10. **Begin**
11. Simulate ← training dataset
12. Construct ROC and locate optimum Euclidean distance (equation 0)
13. Return FNR and FPR for optimum Euclidean distance
14. **End;**
15. Compute gradients and update ANN object weights and biases in direction and step size as per SCG algorithm
16. Train for up to E==1000 epochs (e) using batch-mode offline training
17. *Interrupt for validation every V generations*
18. **Begin**
19. Simulate ← validation dataset for this ensemble member / data fold
20. Construct ROC and locate optimum Euclidean distance (equation 0)
21. IF (early-stopping criterion met OR E==1000) record "best" ANN and exit SCG training algorithm
22. **End;**
23. **End;**
24. On completion of training, simulate with the trained network using the 13th (2012) ensemble evaluation data-fold and store responses together with ROC evaluation metrics
25. Store the trained weights and biases and combined pathway strength matrix in format suitable for ANaNAS
26. **End;**
27. Evaluate overall ROC performance of ensemble using collation of ANN results
28. Evaluate EQR for each input feature; analyse pathway strength vectors over the ensemble and rank the inputs in descending order of EQR
29. **End;**
30. Assess mean and median rank for each input over all ensembles / ANN architectures
31. Repeat once from 2. using reduct of only input features with mean EQR>0
32. Repeat once from 2. using reduct of only the 6 highest mean ranked input features
33. Compare ROC *AuC,* F_{opt} and E_{opt} metrics for the full 12 input features trial with those for the reduct trials using Student's T-test (Fay and Proschan, 2010)

Evaluation of Individual ANN Test Results

After training, each ANN is tested using its assigned "leave-one-out" test data fold. A ROC is constructed by varying the decision threshold between 0 and 1 and the area under the curve (AuC) is evaluated. Due to the small number of samples in the individual ANN test data folds and especially the variable but very small number of water quality failures within the folds, it would not be good practice to attempt comparative evaluations between ANNs using these test results. The ensemble test data fold is provided instead for this purpose. Additionally, the optimal operating points for the ANN are located on its ROC as defined in the subsection: *Optimum Operating Point Location* above. The corresponding optimal threshold values are thus also identified.

A drawback of this approach is that each ANN is likely to require a different optimal threshold, so no single overall optimum is likely to exist for the ensemble. Two approaches to address this are described below.

Evaluation of Ensemble Test Results

Following construction of the entire NFCV ensemble, the ROC curves from all ensemble members are plotted together using the approximately 100 daily samples from the 2012 bathing season as the ensemble test dataset

Two approaches are used to produce the ensemble's combined classification response for each sample:

1. Normalisation of all ANN output spans

   A resultant ensemble ROC curve is produced by normalising all the ANN output ranges[46] then applying a decision threshold value to produce each point on the normalised ensemble majority decision ROC curve. The resultant classification for each threshold value is taken as the class (either "pass" or "fail") that the majority of ensemble members agree

[46] *The spans of output are found to vary from ensemble member to ensemble member, so normalisation allows them to be evaluated against a single set of threshold values to construct the combined ensemble ROC curve.*

upon[47]. Figure 5.10 shows the normalised ANN outputs (y-axis) versus the sample index (x-axis) for the 2012 ensemble test dataset, using Readymoney beach as an example, which comprises an ensemble of seven ANN members.

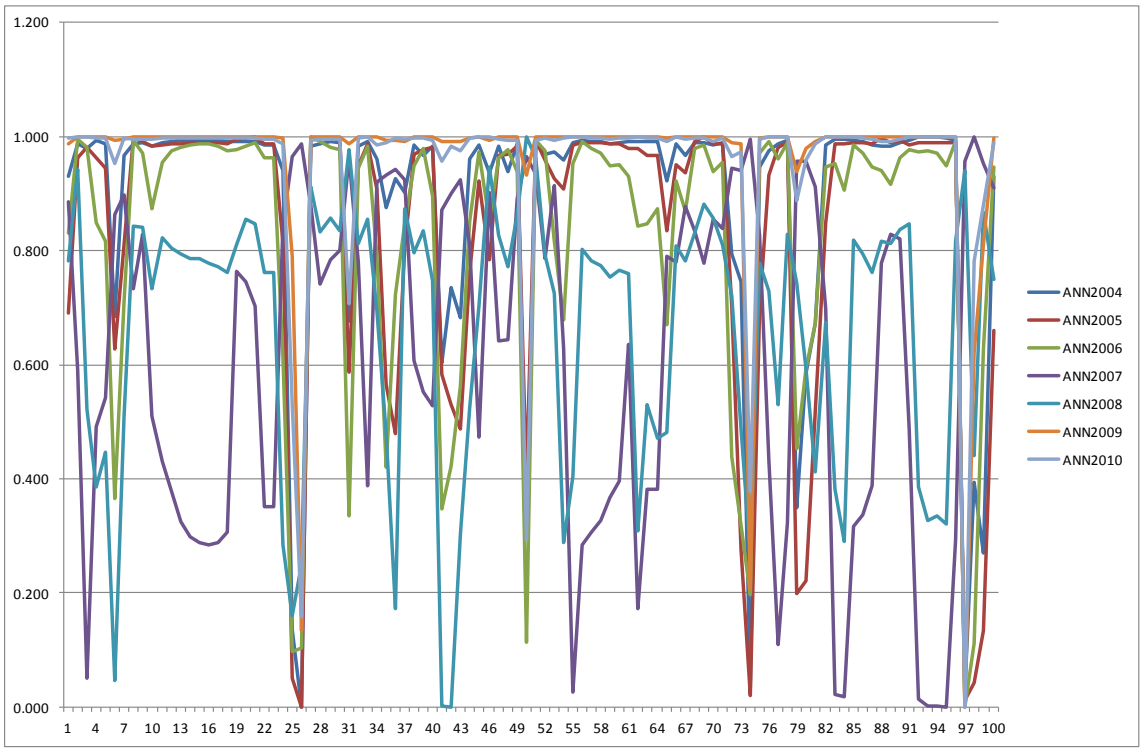


Figure 5.10. Readymoney beach 2012 data fold: Normalised ensemble ANN outputs

The object is to define an optimal threshold (value between 0 and 1) for the ensemble majority decision. By varying the threshold between 1 and 0, the majority decision ROC curve is constructed. The AuC of the majority ROC is then evaluated as well as the location of the optimum operating points, based on F and E_{opt} as described earlier in section 5.2.5.3.

2. Alignment of optimal thresholds for all ANNs

As an alternative to the above, a resultant ensemble ROC curve is produced by rescaling all the ANN output ranges from a datum point of 1.2, so that the rescaled optimum thresholds of the individual ANNs all align at 0.5. A threshold value (between 0 and 1) is applied to produce each point on the ensemble ROC curve. The resultant classification for

[47] *In the event of a tie, the ensemble decision is taken as a "pass", being the class with the larger number of samples and therefore highest probability.*

each threshold value is taken as the class that the majority of ensemble members agree upon. The same metrics as in 1 above are evaluated. Figure 5.11 illustrates the same data as in Figure 5.10, but having been rescaled to align optimum thresholds at 0.5.

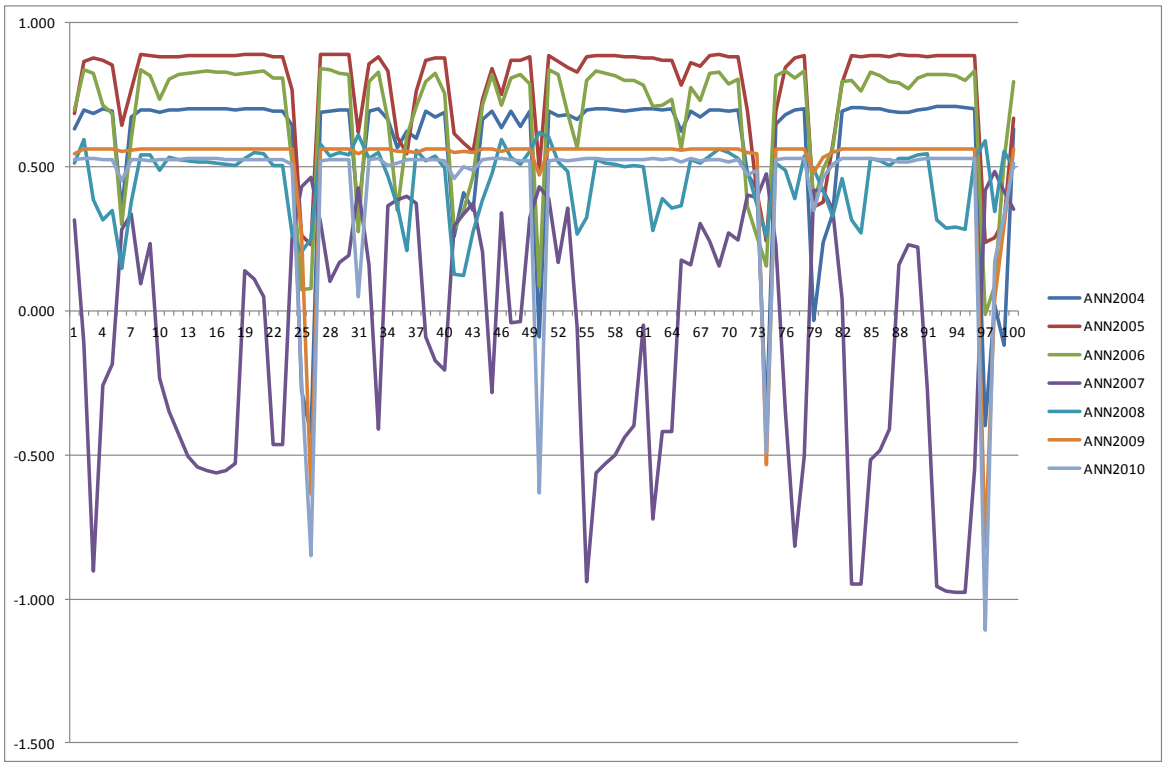


Figure 5.11. Readymoney beach 2012 data fold: Aligned ensemble ANN outputs

Results for the two approaches are then compared.

NPSFS Methodology

The same methodology of Neural Pathway Strength Feature Selection as described in Chapter 4 is applied, by analysing the weights of the ensemble-member ANNs. A reduct of relevant input features is thus constructed and the experiment described above is repeated using just the selected input features. The measure "EQR" – Ensemble interQuartile Range of combined neural pathway strengths for each input feature is again used to evaluate and rank the relevance of input features. An EQR of greater than zero means that at least 75% of ANNs use the given input feature in the same (inhibitory or excitatory) sense. An EQR of less than zero means that between 25% and 75% of ANNs use the input feature in the opposite sense to the remainder of the ANNs.

Two input feature selection strategies are adopted:

1. To include features in the reduct if their EQR is greater than zero. For simplicity, since a number of ensembles with varying ANN architectures are produced in order to test the robustness of the approach, mean EQR values for the entire collection of ensembles are used to determine each input feature's membership of the final reduct.
2. To include exactly half of the original total input features, selected in descending order of EQR value.

The experiment is then repeated using the reduced input feature set for each ANN architecture. The majority AuC of each reduct ensemble is then compared with that of the original ensemble with the full set of input features.

5.3 Results

5.3.1 Decision Tree Models and Simple Trigger Results

The following decision tree and simple threshold results have been produced by Dr Deborah Tyrrell of the Environment Agency and are included as a benchmark for comparison with the ANN results. They are also reported in Duncan et al. (2013d). An ROC approach is not used with these models, so a single operating point is provided by each.

The beach at Seaton (Cornwall) is used for this results section. Table 5.4 shows that the Decision Tree models are capable of predicting both "fail" and "pass" with acceptable misclassification error. There is a total of 10 out of 13 "fails" that are predicted correctly and 79 out of 86 "passes" are also predicted correctly. The models are also tested blindly[48] on data for full bathing seasons (15th May to 30 September) for each year, 2007 to 2011 inclusive. This is in order to count how many exceedances per bathing season would typically be predicted[49].

The number of public advisory days per bathing season ranges from 10 in 2009 and 2011 to 28 in 2008. This is correlated with the very high long-term

[48] *I.e. the data for these years is excluded from the DT training set.*

[49] *Requiring advisory signs to be displayed at the beach, warning the public that is may not be safe to bathe.*

average rainfall recorded in 2008. It is worth noting that this trial uses only weekly compliance samples to evaluate performance, whereas the numbers of advisory signs (not to bathe) per bathing season reported in Table 5.4 are based on daily predictions for the 153 days of each bathing season.

A final set of DT models includes all the data to 2011 in the training set and they are tested using samples from 2012. Five of the eight bathing waters evaluated have samples collected daily (excluding weekends and Bank Holidays) throughout the 2012 bathing season to provide a larger dataset, and hence a more robust model validation than previously available. The daily sampled beaches are Seaton (Cornwall), East Looe, Readymoney, Par, and Porthluney, details of which can be found in Figure 5.2 and Table 5.1, with beach profiles presented in Appendix A.

Table 5.4. Decision Tree Validation Results for Seaton (Cornwall)

Data Used to Build Model	Samples Used to Build Model	Test Year	Samples In Test Year	Sample Exceedances in Test Year	Predicted Pass, Sample Pass (PP)	Predicted Fail, Sample Pass (FP)	Predicted Fail, Sample Fail (FF)	Predicted Pass, Sample Fail (PF)	Total Signs in Bathing Season (153 days)	Model Predictor Variables
2000-06	140	2007	20	1	16	3	0	1	17	Rainfall only (24, 48, 72, 96, and 120hr)
2000-07	160	2008	20	7	10	3	5	2	28	Rainfall only (24, 48, 72, 96, and 120hr)
2000-08	180	2009	19	1	18	0	1	0	10	Rainfall only (24, 48, 72, and 96hr)
2000-09	200	2010	20	3	17	0	3	0	11	Rainfall only (24, 48, 72, and 96hr)
2000-10	220	2011	20	1	18	1	1	0	10	Rainfall only (24, 48, 72, and 96hr)

5.3.1.1 Comparison with Simple Trigger

Using the same 2012 dataset, a simplified assessment is included using single 24hr rainfall triggers of 0, 5, 10 and 15mm, and these results are compared with those of the DT models. The results from these are presented in Figure 5.12 and Table 5.5.

The decision tree models (red diamonds) give a similar level of accuracy to that produced using a single 24hr rainfall trigger of 10mm (green squares)

shown in Figure 5.12. This is significant because the decision tree method relies on having a reasonable number (typically >10%) of data samples exceeding the bacteriological threshold[50] in order to train the model[51], whereas a single 24hr rainfall trigger may be applied to any number of bathing waters of varying quality. It is worth noting, however, that the calibration set-points of the simple trigger thresholds will vary from beach to beach. These can be determined for example by choosing a threshold corresponding to a maximum value of modified F-measure (equation 5.3).

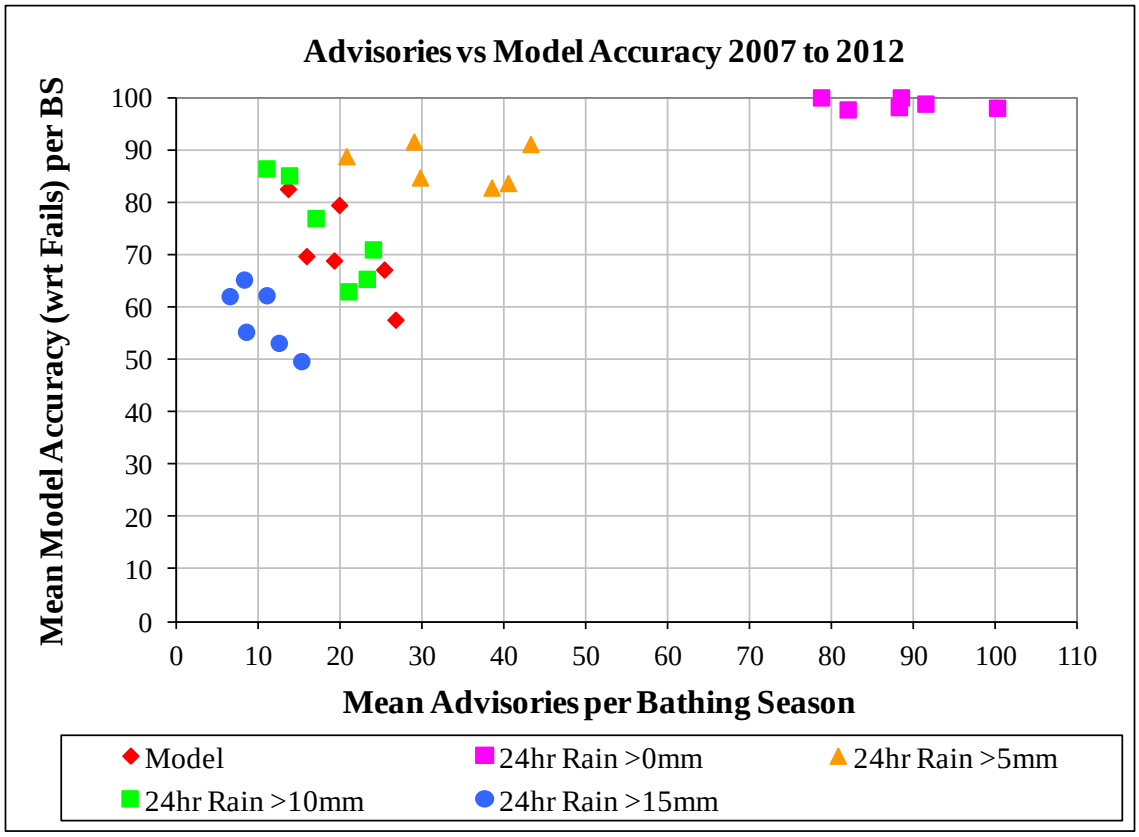


Figure 5.12. Comparisons of Decision Tree Models with Simple Triggers – Advisories vs. Model Accuracy 2007-2012

Figure 5.12 also shows that a trade-off exists between threshold used for the simple trigger and the number of advisories that would have to be issued in a bathing season. As the rainfall trigger decreases, the number of "fails" correctly predicted increases, and so does the potential number of public advisories. This is important because to implement an operational bathing water warning system, the rainfall triggers set would have to be determined not only

[50] i.e. in the "fail" class
[51] This would only be expected for the bathing waters with poorer water quality.

by the absolute accuracy of the predictions, but also by recognising the number of public advisories deemed acceptable by beach managers.

By using the ROC approach, it is possible to construct ROC curves for the simple trigger models, by varying the rainfall thresholds, from zero to a figure above the maximum antecedent rainfall recorded. Figure 5.13 shows such a curve for Porthluney beach constructed using all data samples 2000-2012 inclusive. Area under the curve (AuC) is 0.797. Figure 5.14 illustrates the same, but with the dry weather failure (DWF) data samples removed. This has the effect of increasing the AuC to 0.868. It can be noted that these curves both most closely approach the optimum [0, 1] point at a threshold of 3mm.

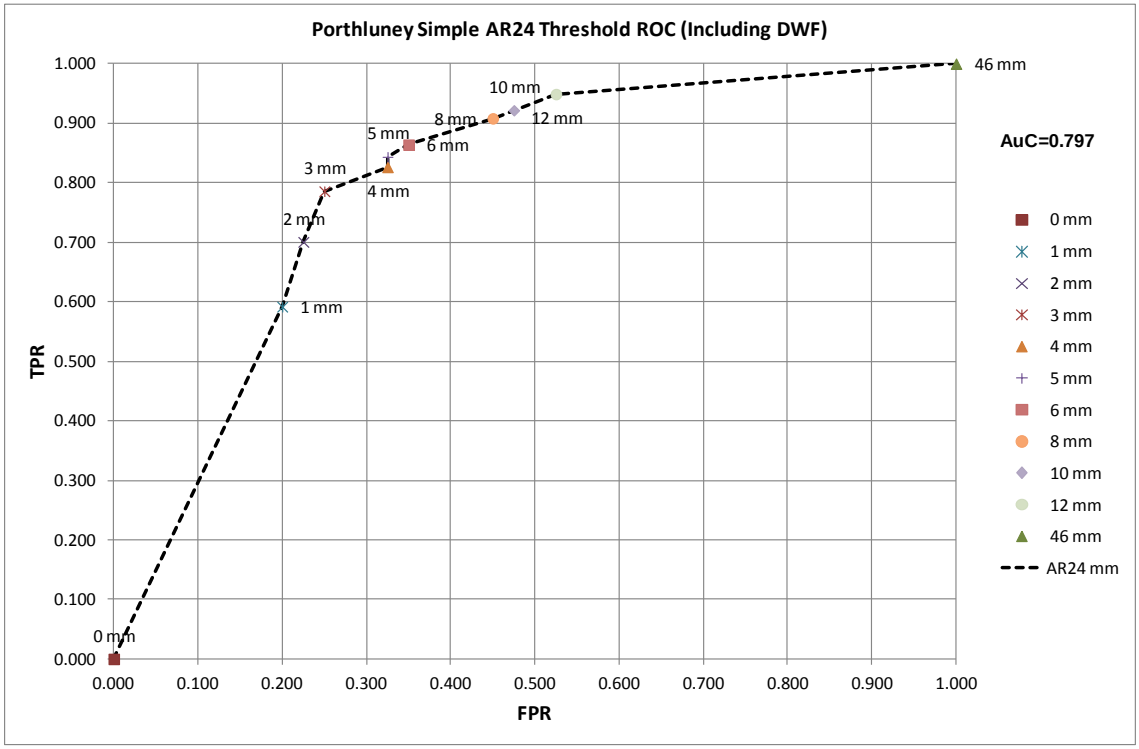


Figure 5.13. Porthluney Simple AR24 Threshold ROC (Including DWF)

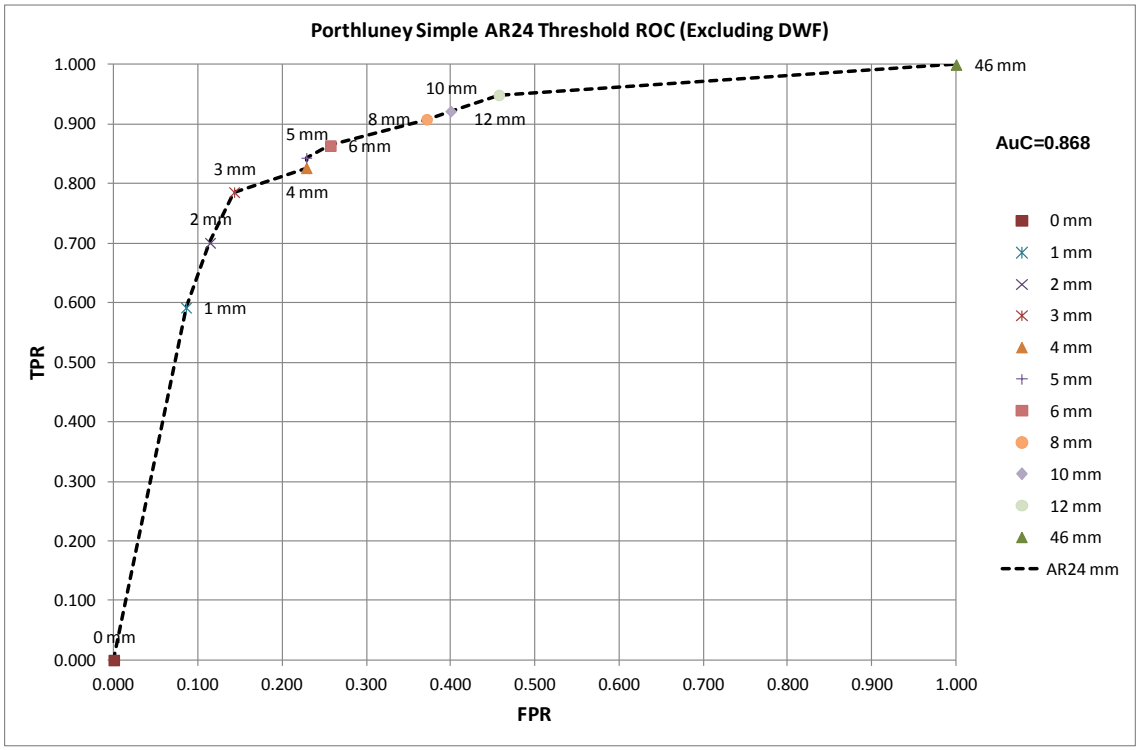


Figure 5.14. Porthluney Simple AR24 Threshold ROC (Excluding DWF)

Alternative Simple Trigger based on Salinity

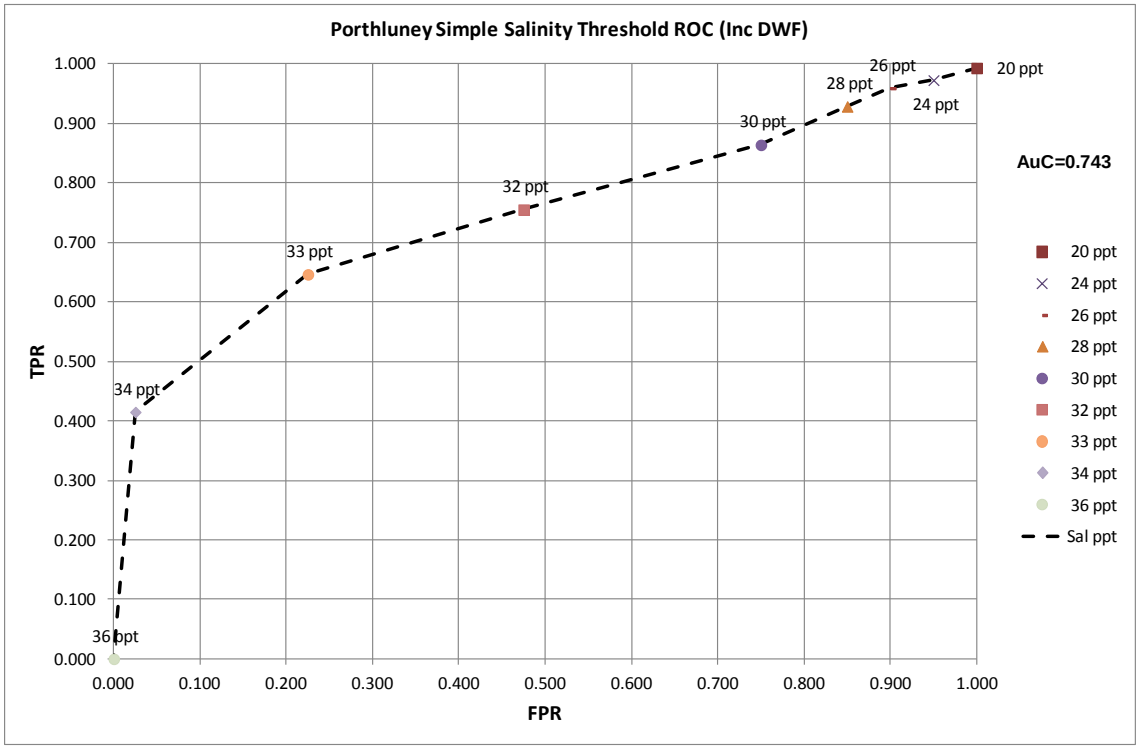


Figure 5.15. Porthluney Simple Salinity Threshold ROC (Including DWF)

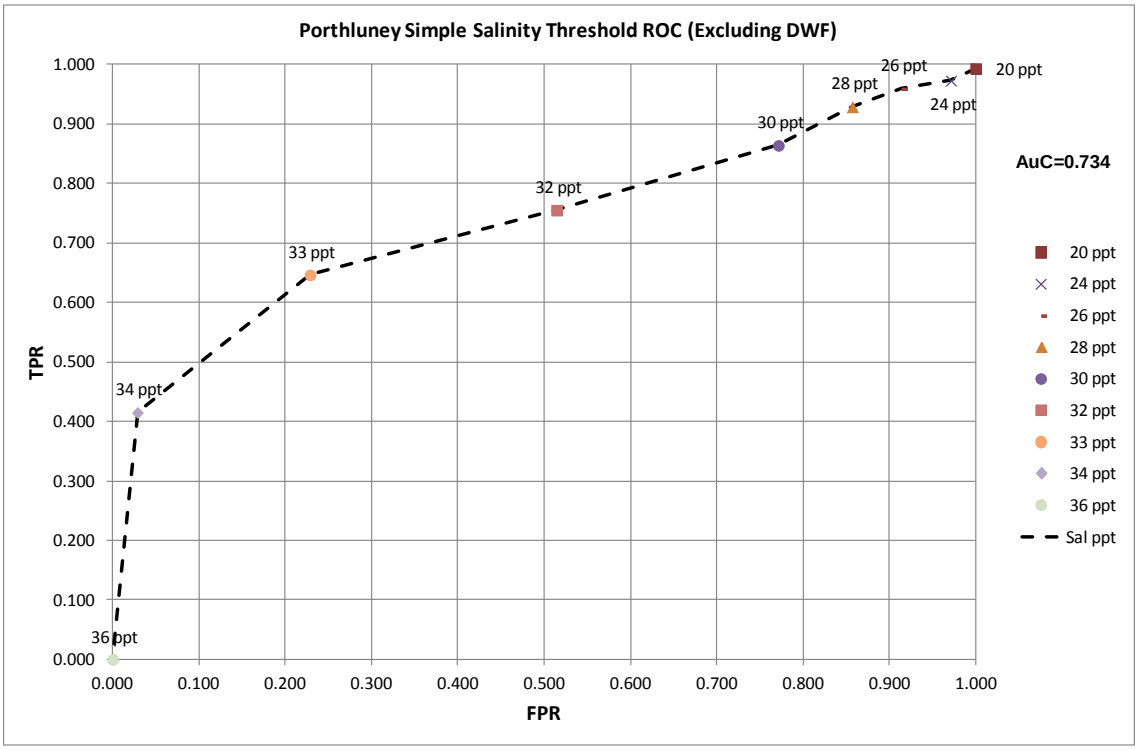


Figure 5.16. Porthluney Simple Salinity Threshold ROC (Excluding DWF)

A similar pair of simple threshold ROC curves for bathing water sample salinity is also constructed in Figure 5.15 and Figure 5.16. These are included here because, as will be seen in the NPSFS results section, Salinity and AR24 are the two input features consistently ranked most relevant, based on EQR.

It is interesting to observe that, whilst the AuC for the ROC including DWF is 0.743, AuC for the curve with the dry weather failures (DWF) removed is marginally reduced to 0.734. It is not suggested to use salinity as a simple trigger as these figures of AuC are probably not adequate. However, salinity is a parameter that can be measured readily and speedily and its use could potentially provide robustness and improved performance for the simple trigger models in the presence of dry weather failures. This conclusion however would need further experimental validation.

The physical interpretation of this is that salinity of a solution is correlated with bactericidal activity as are the disinfecting properties of both sodium and chlorine. The mean salinity for the passing samples is 32.8ppt, whereas the mean for the failing samples (including DWF) is 30.9ppt.

5.3.1.2 Discussion of DT and Simple Trigger Results

Table 5.5 also shows the results of additional fine tuning of the single 24hr or 48hr rainfall trigger levels (mm)[52]. Also presented is an assessment of how the rBWD classification for 2012 would be improved by being able to discount samples that were correctly predicted to exceed the bacteriological threshold. Classifications are colour-coded for convenience. It is clear from this that the rBWD classification can be improved with discounting, and that both methods are applicable.

Table 5.5. DT and Simple Threshold Prediction Results for 2012

Bathing Water	Sample Fails	Dry Weather Fail	Samples without Discount	rBWD 2012 without Discount	24hr Rain Trigger	48hr Rain Trigger	%age Correct Predictions	Samples with Discount	Samples Discounted	rBWD 2012 with Discount	Advisories 2010	Advisories 2011	Advisories 2012	Total Advisories 2010 to 2012	Method
Mothecombe	15	0	80	Poor	10		**90**	70	10	Good	13	18	23	54	**Simple trigger**
							84	73	7	Sufficient	7	12	33	**52**	Decision tree
Seaton (Cornwall)	9	0	80	Poor	8.5		**96**	72	8	Good	15	11	22	48	Simple trigger
							94	74	6	Sufficient	11	10	12	**33**	**Decision tree**
East Looe	13	4	80	Poor		19	84	75	5	Poor	17	8	25	**50**	Simple trigger
							86	75	5	Poor	12	12	30	54	**Decision tree**
Readymoney	8	0	80	Suff – icient	10		89	75	5	Good	11	7	22	40	Simple trigger
							90	76	4	Good	8	13	18	**39**	**Decision tree**
Par	6	0	80	Suff – icient	15		**94**	77	3	Good	10	3	11	**24**	**Simple trigger**
							94	78	2	Sufficient	9	6	19	34	Decision tree
Porthluney	15	2	80	Poor	7.7		**90**	70	10	Sufficient	13	8	23	**44**	**Simple trigger**
							86	72	8	Sufficient	13	15	26	54	Decision tree
Ilfracombe Wildersmouth	22	3	80	Poor		9	66	70	10	Poor	34	45	61	140	Simple trigger
							70	69	11	Poor	34	38	40	**112**	**Decision tree**
Burnham Jetty	9	0	80	Poor	10		89	76	4	Poor	3	6	14	**23**	Simple trigger
							93	76	4	Sufficient	2	14	25	41	**Decision tree**

The conclusion whether to use single rainfall triggers or decision trees is a balance between the accuracy of the predictions, the number of advisories, and whether rBWD class change is achieved. The simple rainfall trigger method is considered appropriate for Mothecombe, Par, and Porthluney. The decision tree method is considered appropriate for Seaton (Cornwall), East Looe,

[52] *East Looe and Ilfracombe Wildersmouth beaches are modelled using a 48-hour antecedent rainfall trigger rather than the 24-hour trigger used elsewhere as this is found to be more accurate.*

Readymoney, Ilfracombe (Wildersmouth), and Burnham Jetty. For East Looe and Ilfracombe (Wildersmouth), further work will be required to improve the water quality and reduce the number of times water quality is impacted in dry weather, so that rBWD class change through discounting can be achieved.

5.3.2 ANN Model Performance Results

Results presented in this section are for the ANN trials for the five beaches (indicated in Table 5.1), for which daily sampling data for the 2012 bathing season is available. A set of observations of sufficient number is required for the evaluation of overall performance of the ensembles using an ROC scenario and this is provided by the 100 samples taken at each of the 5 beaches in 2012. Use of a set of 20 weekly compliance samples results in poor resolution for the ROC curves, due to the very low number of negative ("fail") samples, so results for the seven weekly-sampled beaches in 2012 are not included.

5.3.2.1 NFCV Ensemble Performance

Detailed results are presented here for ANN ensembles using the full set of 12 input features for Seaton (Cornwall) and Porthluney beaches. Summary results are then presented for all 5 beaches, using the metric of ROC area under the ensemble normalised majority decision curve (*AuC*) and the two measures of maximum F-measure (equation 0) (*F*) and minimum Euclidean distance (equation 0) (E_{opt}) taken as alternative optimum operating points from the above ROC curve. Datasets exclude dry weather failures (DWF) unless otherwise stated.

Figure 5.17 displays the full set of ROC curves for Seaton beach for the best ensemble of ANNs with 27 hidden units each, using the SCG algorithm (3.4) with ROC AuC as single objective function. There are a set of 12 individual ROCs for the ANN ensemble members (ANN2000-ANN2011) named after the individual test data folds (bathing seasons) used for each.

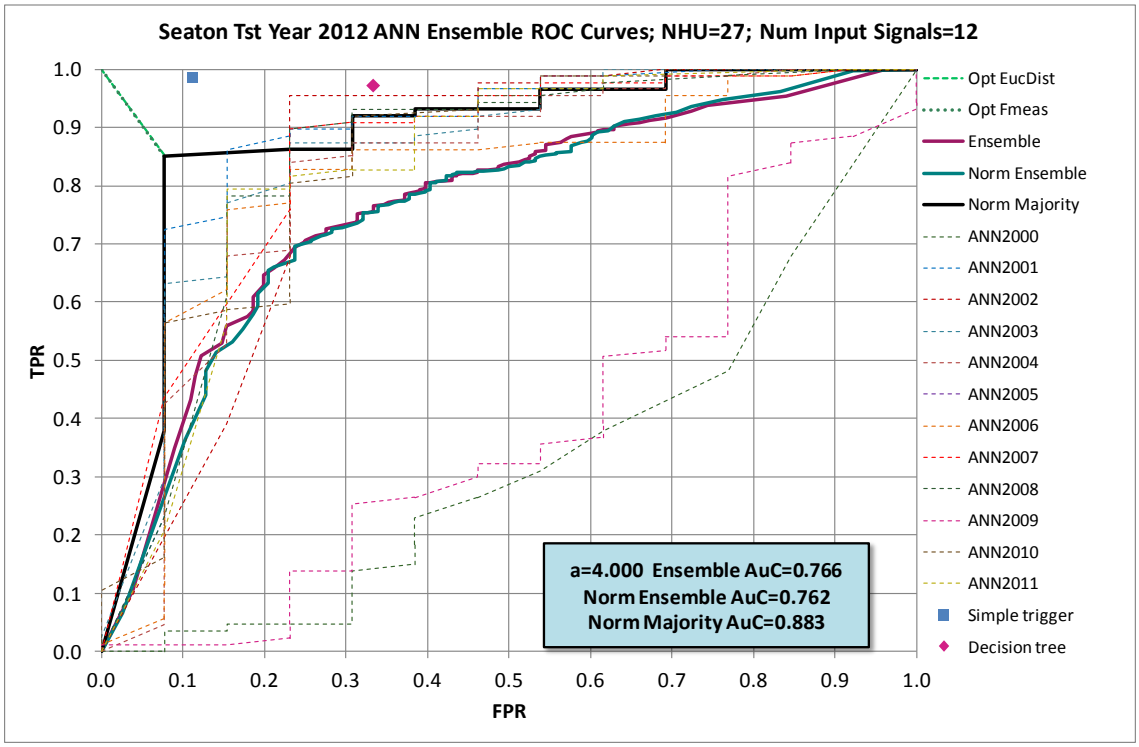


Figure 5.17. Seaton Test Year 2012 ANN Ensemble ROC Curves; NHU=27; Num Input Signals=12

There are then 3 ensemble ROCs for:

1. The Ensemble mean response (magenta) – this is produced based on the raw outputs of the 12 ANNs and applying a variable threshold to all these outputs simultaneously.
2. The Normalised Ensemble mean response (cyan) – this is produced based on the normalised outputs of the 12 ANNs (as described in section 5.2.5.4 "*Evaluation of Ensemble Test Results*" and applying a variable threshold to all these outputs simultaneously.
3. The Normalised Majority response (black) – this is produced based on the normalised outputs of the 12 ANNs (as described in section 5.2.5.4 "*Evaluation of Ensemble Test Results*" and applying a variable threshold to all these outputs simultaneously and taking the Pass/Fail decision of the majority of ensemble members.

The areas under the curve (AuC) of the 3 ensemble ROCs are shown in the blue annotation textbox. The fact that the ensemble mean and normalised ensemble mean curves are almost coincident shows that the inclusion of the ANN output span term in the training objective function is working well and most outputs are approaching the unity span. As expected, the ensemble majority decision is much better than the mean ROC, so the normalised ensemble majority decision ROC will be used as the standard decision-making mechanism for ensembles and the standard evaluation metric henceforward. It can be noted that there are two ANNs (2000 and 2009) that are outliers with poor performance, but the ensemble approach is tolerant of this and the ensemble majority ROC is not affected. However, the ensemble-mean ROCs are both affected; a further reason not to use these.

The locus/loci on the ensemble majority ROC taken for the F and E_{opt} metrics is indicated by the green dotted and dashed lines, which, here, are coincident but may not always be. These loci are influenced by the value of 'a' used; here 4.0. The value of 4 has the effect of stretching the x-axis by a factor of 4, meaning that optimal operating points on the ROCs tend to be selected towards the left of the curve. This is discussed and shown in Figure 5.6 and Figure 5.7.

Also shown in Figure 5.17 for comparison are the operating points for the Environment Agency DT (magenta diamond) and simple threshold (blue square) models. It can be seen that these both are above the ensemble majority ROC but, due to the use of 'a'=4 throughout, the E_{opt} figures for these are worse (higher) than for the ANN ensemble. F is also worse (lower) for the DT models than for the ANN. These are recorded in Figure 5.19 and Figure 5.20. These operating points set the false alarm rate (FNR) very low indeed, but at the cost of tolerating a high missed alarm rate (FPR).

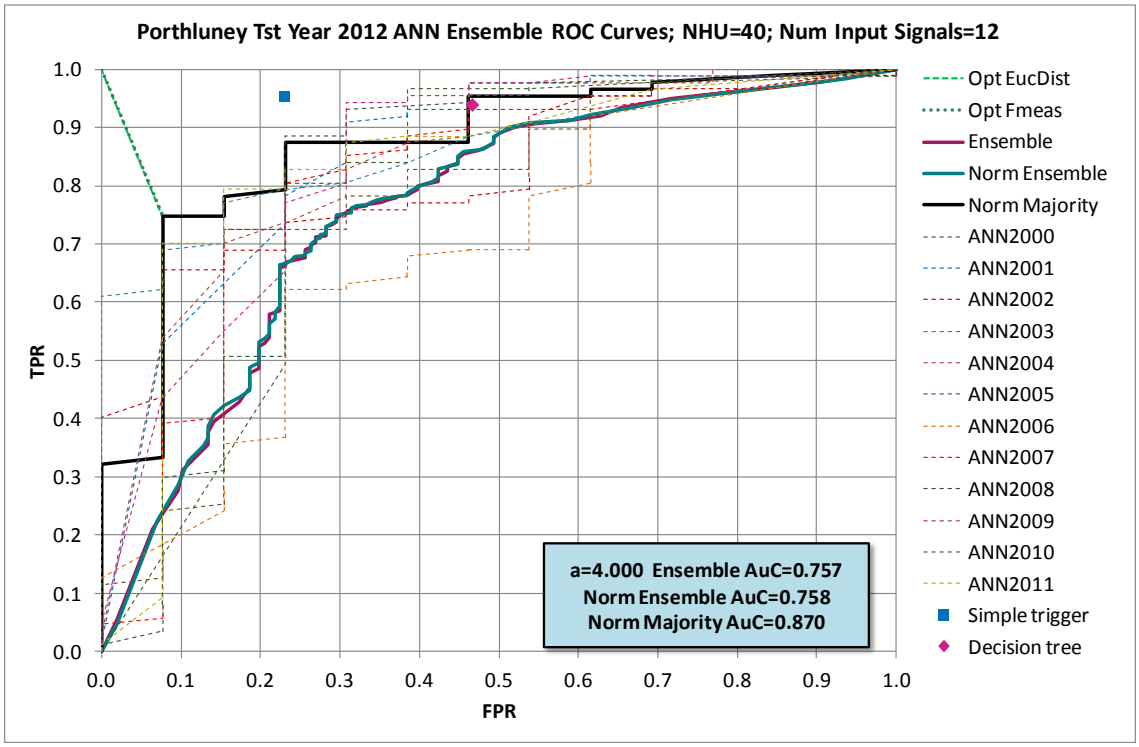


Figure 5.18. Porthluney Test Year 2012 ANN Ensemble ROC Curves; NHU=40; Num Input Signals=12

Figure 5.18 illustrates the similar ROC for Porthluney beach. Here, the optimal architecture was found to be with 40 hidden units. Area under the ensemble majority ROC is very similar: (Seaton = 0.883; Porthluney = 0.870). Here the DT and simple threshold operating points are further towards tolerating missed alarms, with the DT model having FPR=0.467.

Table 5.6 presents a comparative summary of model performance results for all 5 beaches. Because the DT and simple threshold results are based on a single operating point, the two metrics of F and E_{opt} are used to compare all models. The AuC metric is presented separately, for the ANN models only. Also shown are the TPR, FPR, TNR and FNR figures for the operating points of the DT and simple threshold models and the optimum operating points for the ANN models. Figure 5.19 shows values of F-measure for all 5 beaches for each of the three categories of model. Here, the ANN ensemble model results are those of the SCG/ROC AuC trained ensembles. In each case, the best-performing architecture (number of hidden units) is chosen.

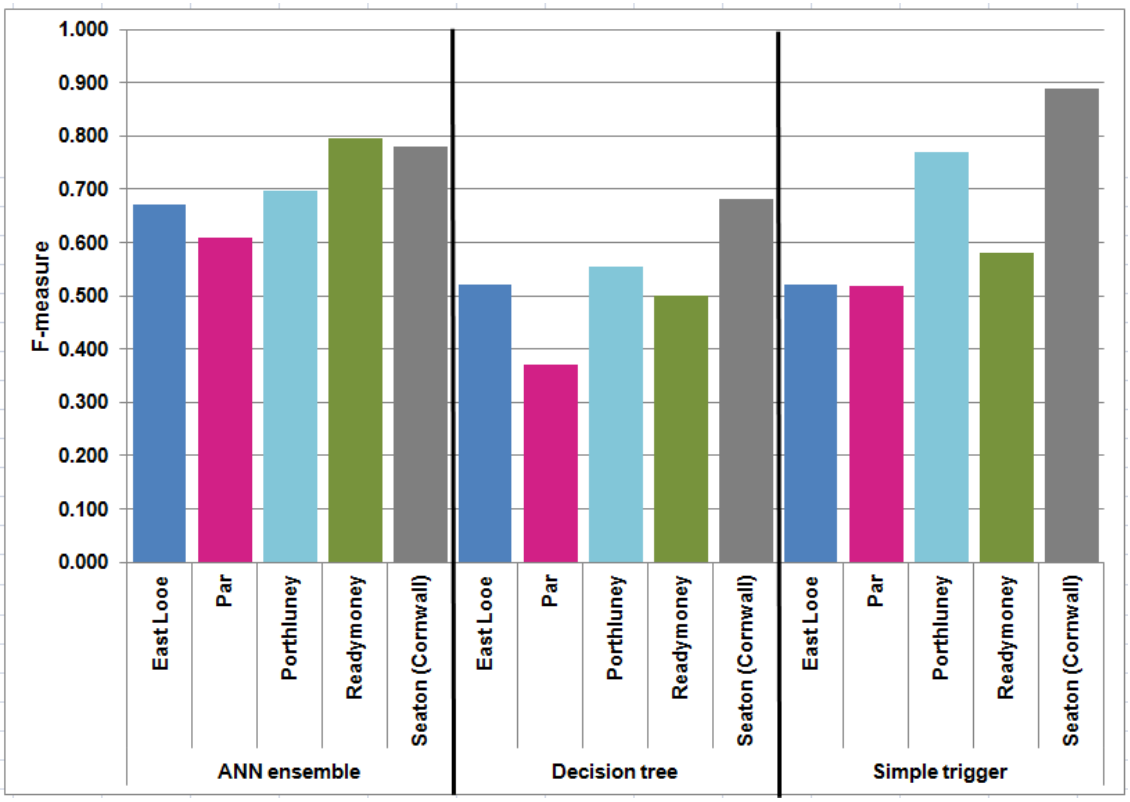


Figure 5.19. Comparison of F-measure for all beaches and models

It can be seen that, apart from the simple threshold models for Seaton and Porthluney, the ANN models produce the best (highest) performance for the F-measure metric.

Table 5.6. Summary of results for 5 beaches for all models

Beach	Trig AR24 (mm)	Trig AR48 (mm)	PP True positives	PF False positives	FF True negatives	FP False negatives	TPR True positive rate	FPR False positive rate	TNR True negative rate	FNR False negative rate	F-measure	Euclidean Distance	Method	Training Algorithm / Metric
Seaton (Cornwall)	10		70	1	8	1	**0.986**	0.111	0.889	**0.014**	**0.889**	0.445	Simple trigger	
			69	3	6	2	0.972	0.333	0.667	0.028	0.682	1.334	Decision tree	
							0.851	**0.077**	**0.923**	0.149	0.779	**0.342**	**ANN ensemble**	**SCG / ROC AuC**
							0.793	0.154	0.846	0.207	0.679	0.534	ANN ensemble	NSGA2 / FPR_FNR
East Looe	15	15	60	4	5	7	0.896	0.444	0.556	0.104	0.521	1.781	Simple trigger	
			64	5	5	3	**0.955**	0.500	0.500	**0.045**	0.521	2.001	Decision tree	
							0.000	**0.000**	**1.000**	1.000	**0.671**	**1.000**	**ANN ensemble**	**SCG / ROC AuC**
Readymoney	10		66	3	5	6	0.917	0.375	0.625	0.083	0.581	1.502	Simple trigger	
			68	4	4	4	**0.944**	0.500	0.500	**0.056**	0.500	2.001	Decision tree	
							0.903	**0.000**	**1.000**	0.097	**0.796**	**0.097**	**ANN ensemble**	**SCG / ROC AuC**
Par	15		72	3	3	2	0.973	0.500	0.500	0.027	0.517	2.000	Simple trigger	
			73	4	2	1	**0.986**	0.667	0.333	**0.014**	0.370	2.667	Decision tree	
							0.727	**0.167**	**0.833**	0.273	**0.610**	**0.720**	**ANN ensemble**	**SCG / ROC AuC**
Porthluney	8		62	3	10	3	**0.954**	0.231	0.769	**0.046**	**0.769**	0.924	Simple trigger	
			61	7	8	4	0.938	0.467	0.533	0.062	0.556	1.868	Decision tree	
							0.747	**0.077**	**0.923**	0.253	0.698	0.398	ANN ensemble	SCG / ROC AuC
							0.885	0.154	0.846	0.115	0.753	**0.276**	**ANN ensemble**	**NSGA2 / FPR_FNR**

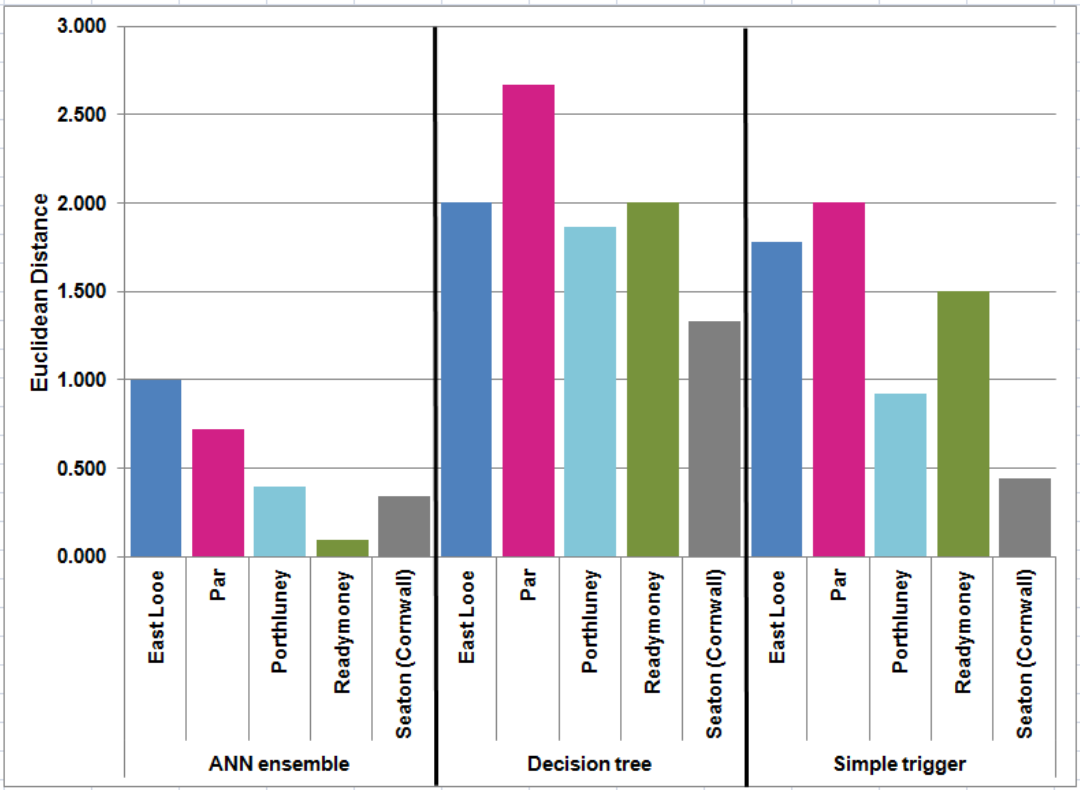


Figure 5.20. Comparison of Optimum Euclidean Distance (E_{opt}) for all beaches and models

Figure 5.20 demonstrates the best (lowest) performance for every beach for the ANN SCG/ROC AuC ensemble models, based on the Euclidean Distance metric. These results all use an 'a' value of 4.0.

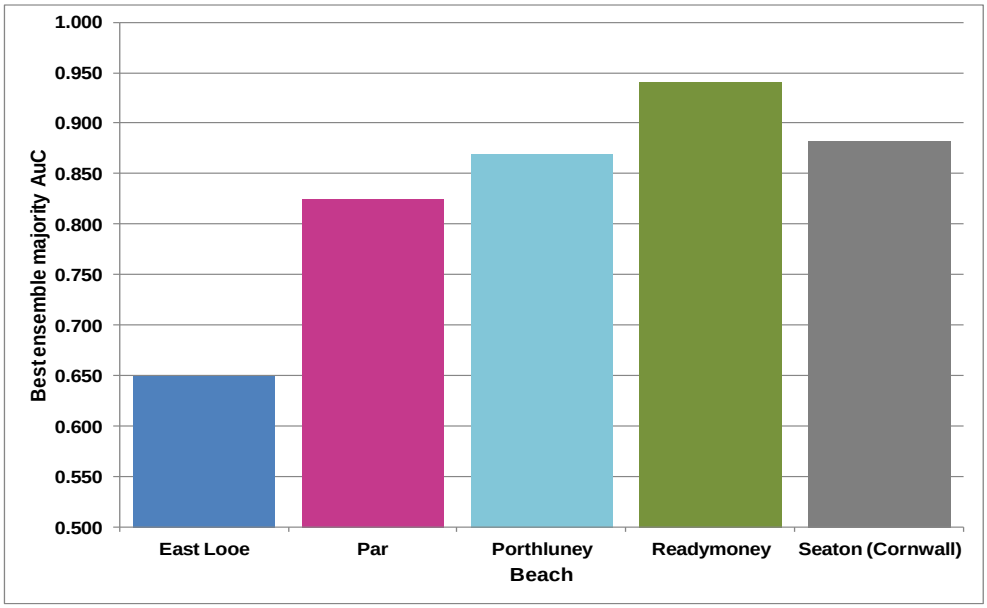


Figure 5.21. Comparison of area under the ROC curve (AuC) for ANN ensembles for all beaches

Figure 5.21 compares the ANN (SCG/ROC AuC – trained) ensemble best models' performance using the ROC AuC metric. In each case, the best-performing architecture (number of hidden units) is chosen, which may be different from beach-to-beach. Apart from for East Looe, all models perform well. In the case of Par and Readymoney, only 8 of the 12-input features are available in the dataset, so these are used to produce these results. For East Looe, the performance of AuC=0.65 is somewhat better than random guessing, but is significantly lower than for the other beaches. It is suspected that for 2012, either there is an extraneous causal factor, not covered by the 12-input features included in the model, that leads to the water quality failures; or there is an error in the preparation of the data. The experiments are therefore repeated for East Looe only, using the 85 daily samples from the 2013 bathing season and the best AuC result of 0.806 is much more in line with the 2012 results for the other beaches.

5.3.2.2 Comparison of normalised and aligned ROC curve performance

Section 5.2.5.4 "Evaluation of Ensemble Test Results" discusses the two alternative approaches used to combine the outputs from the individual ANN ensemble members: normalisation and threshold alignment. This section presents the performance results associated with these, using Seaton (Cornwall) beach models to demonstrate.

Figure 5.22 illustrates the aligned ensemble and aligned majority decision ROC curves for comparison with Figure 5.17, which shows the equivalent using normalisation of the ANN outputs. It can be seen that, whilst the AuC for the normalised ensemble is 0.883, that of the aligned ensemble is 0.881 – an insignificant difference. However, the aligned ROC appears rounded-off in shape and so loses the advantage of a close approach to the ideal point of [TPR=1, FPR=0]. This results in the following (Table 5.7):

Table 5.7. Comparison of metrics F and E_{opt} for aligned and normalised ensembles

Seaton (Cornwall) a = 4.0	**F-measure**	**Euclidean distance**
Aligned ensemble majority	.679	.421
Normalised ensemble majority	**.779**	**.342**

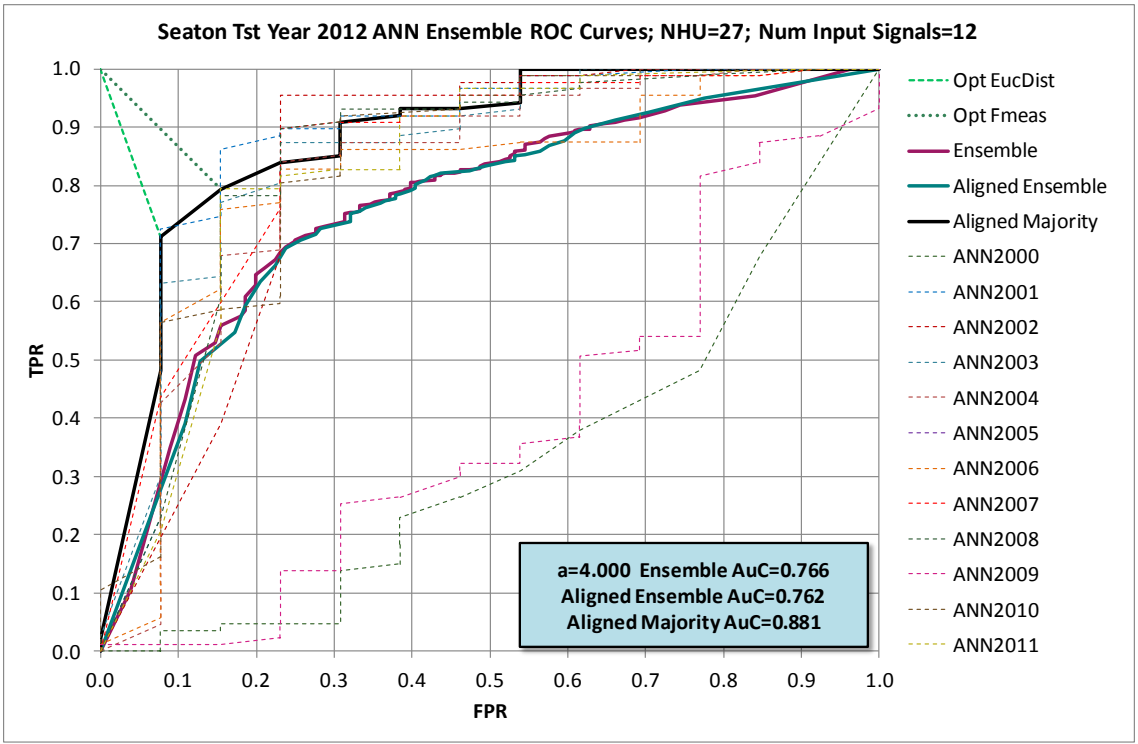


Figure 5.22. Seaton Test Year 2012 ANN Aligned Ensemble ROC Curves

Table 5.7 shows best performance for both metrics, *F* (maximum) and E_{opt} (minimum) for the normalised ensemble majority decision ROC. This is typical of results obtained for other ANN architectures and bathing waters.

Exceptionally, the alignment technique can demonstrate improved results for the 3 metrics: *AuC*, *F* and E_{opt}, when compared with normalisation. However, the normalisation technique is found to perform most consistently; so is adopted throughout unless otherwise stated. Also, due to the inclusion of the ANN output span term in the single-objective optimiser for ANN training, the normalisation process is frequently making only minor adjustments to the ANN output responses – as is demonstrated by the coincidence of the "Ensemble" and "Norm ensemble" ROC curves in Figure 5.17 and Figure 5.22.

5.3.2.3 Results from NSGA-II based training of ANNs

This section presents results for the NSGA-II (evolutionary algorithm) based training of ANNs, using false positive rate (FPR) and false negative rate (FNR) as two objective costs that exist in a trade-off with each other, but both need to be minimised to achieve the optimal model. The NSGA-II objective function constructs an ROC for each candidate solution member of the evolutionary population and uses equation 0 with a=1 to find the optimal operating point on the ROC. From this, the values of FPR and FNR for this point are extracted for use as the two objective costs for the candidate.

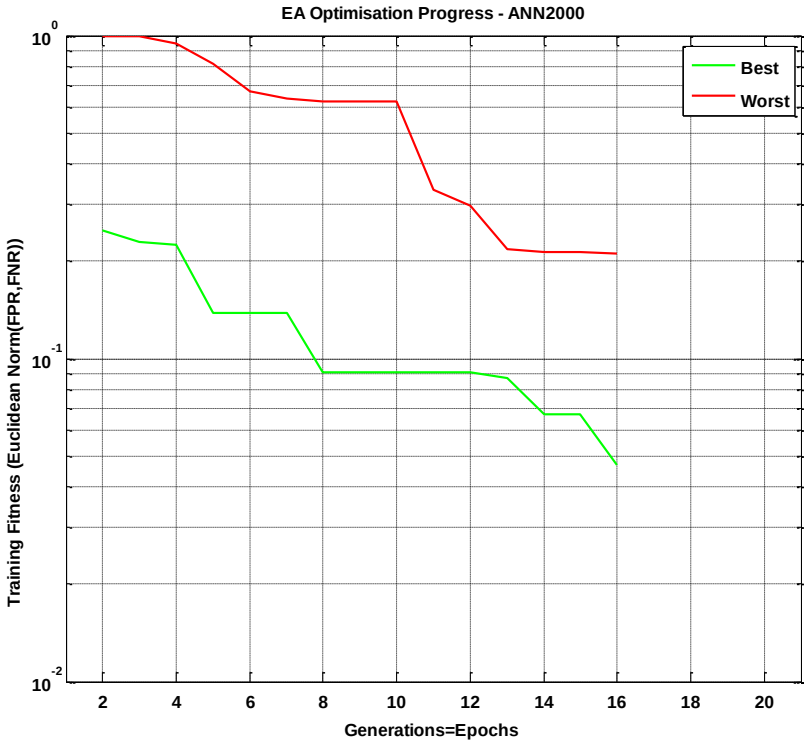


Figure 5.23. Typical NSGA-II progress of population training error fitness during training

Figure 5.23 illustrates typical progress of population training error fitness by evolutionary generation (equivalent to training epoch). The example is taken from the ANN using the 2000 bathing-season as test fold, for the Seaton Cornwall beach model. The green trace shows the fitness of the "best" population member at each generation, as measured by ||FPR, FNR||, where ||x|| is the Euclidean Norm of x. The red trace shows the same for the "worst" population member – thus showing the spread of training error fitnesses for the entire population between the two traces. It is found that convergence to

optimum is rapid and training seldom needs to progress beyond 30-40 generations before one or more of the early stopping criteria are met; as in this example, where optimum validation error is reached after 16 generations. Progress of training is dependent on the values of P_c (probability of crossover per base-pair) and P_m (probability of mutation per location on the child chromosome) set; but a good overall compromise for the lengths of chromosomes found for the trialed architectures of between 5 and 40 hidden units is $P_c = 0.2$ and $P_m = 0.1$.

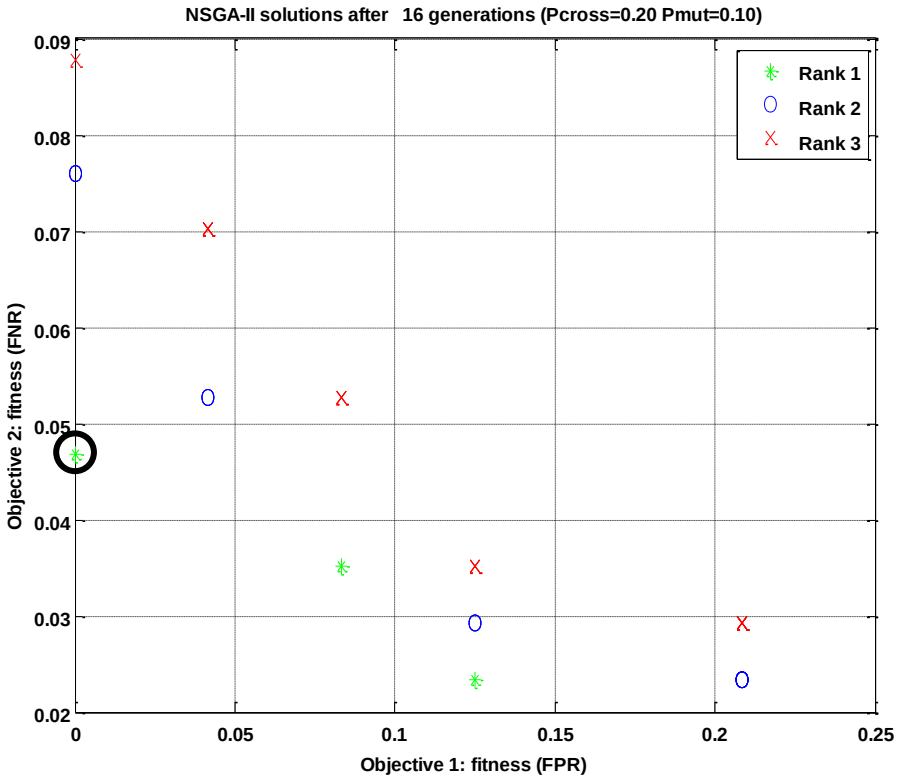


Figure 5.24. 2D Pareto Fronts of solutions for NSGA-II after 16 generations

Figure 5.24 shows the values of fitness for population candidate solutions in the 2-objectives (FPR and FNR) on completion of the same training run as for Figure 5.23. Values of fitness are evaluated for the training dataset for the year 2000 NFCV fold[53]. Solutions are classified into rank 1, 2 and 3 Pareto fronts, and coloured green, blue and red. The solution selected to represent the 2000 data fold in the NFCV ensemble is the rank 1 (non-dominated) one with training fitness meeting equation 0. This is circled in the figure.

[53] *This includes years 2002 to 2011 (year 2001 is the validation fold).*

5.3.2.4 Comparison of performance based on training algorithm used

This section presents comparative results for the NSGA-II (evolutionary) and SCG (Scaled Conjugate Gradients) algorithms. This is made over a range of ANN architectures with 5, 8, 12, 18, 27 and 40 hidden units for the Seaton (Cornwall) beach. The three metrics of *AuC*, *F* and E_{opt} are used for comparison. *AuC* and *F* should be maximised, whereas E_{opt} should be minimised.

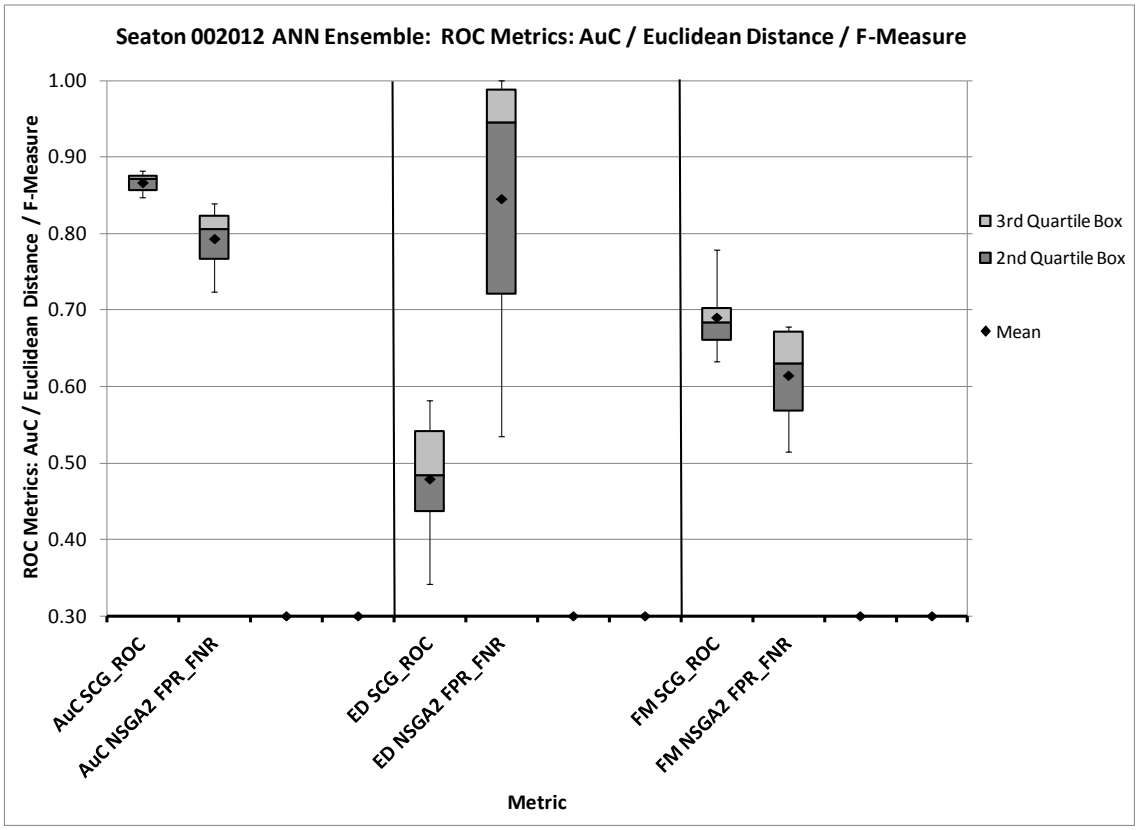


Figure 5.25. Comparison of SCG and NSGA-II ANN training algorithm performance for Seaton

For Seaton (Cornwall) beach, Figure 5.25 shows the spread of values over the range of ANN architectures in each box-and-whisker for the 3 metrics: Normalised ensemble majority *AuC* (left section), optimum Euclidean distance (ED) (centre section) and F-measure (FM) (right section). Within each section, SCG algorithm performance is on the left and NSGA-II is on the right. Inspection reveals that in each case, SCG performs better.

Table 5.8.Seaton T-test results comparing SCG and NSGA-II performance

2-tailed paired T	0.018		0.013		0.124	
NHU	AuC SCG_ROC	AuC NSGA2 FPR_FNR	ED SCG_ROC	ED NSGA2 FPR_FNR	FM SCG_ROC	FM NSGA2 FPR_FNR
5	0.876	0.839	0.462	0.534	0.705	0.679
8	0.847	0.827	0.553	0.649	0.658	0.679
12	0.866	0.814	0.429	0.939	0.672	0.649
18	0.853	0.798	0.582	0.951	0.633	0.610
27	0.883	0.757	0.342	1.000	0.779	0.556
40	0.875	0.725	0.506	1.000	0.696	0.515

The results in the top row of Table 5.8 compare the results for SCG and NSGA-II optimised ANN ensembles and show the probabilities that the compared metrics of AuC (left), Eopt (centre) and F (right) are from the same population. This shows that in the case of AuC and Eopt, SCG performs better than NSGA-II with a greater than 95% significance level, whereas the results for F are inconclusive.

The opposite is however observed for Porthluney as illustrated in Figure 5.26. This shows NSGA-II performing marginally better.

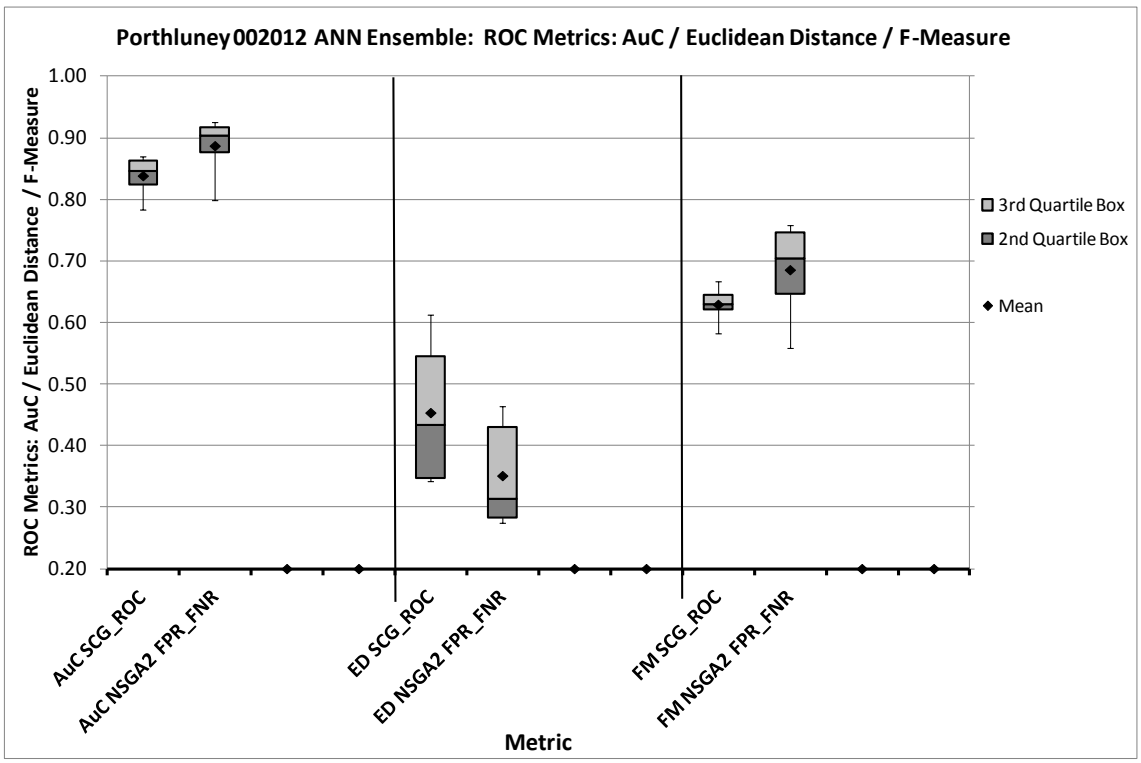


Figure 5.26. Comparison of SCG and NSGA-II ANN training algorithm performance for Porthluney

Table 5.9. Porthluney T-test results comparing SCG and NSGA-II performance

2-tailed paired T	0.087		0.158		0.074	
NHU	AuC SCG_ROC	AuC NSGA2 FPR_FNR	ED SCG_ROC	ED NSGA2 FPR_FNR	FM SCG_ROC	FM NSGA2 FPR_FNR
5	0.867	0.869	0.351	0.464	0.649	0.638
10	0.818	0.799	0.556	0.464	0.582	0.558
15	0.842	0.925	0.345	0.276	0.634	0.733
20	0.784	0.921	0.612	0.275	0.625	0.750
30	0.850	0.900	0.514	0.326	0.620	0.676
40	0.870	0.908	0.342	0.302	0.667	0.759

The results of T-tests documented in Table 5.9 show that there is a less than 95% significance level that the populations for the two algorithms differ, for all three metrics.

Based on these fairly inconclusive tests it is decided to present the results for the SCG single-objective AuC performance function for comparison for all 5 beaches. However, results for all 5 using NSGA-II could also potentially be presented. The results using SCG are presented in Table 5.6.

5.3.2.5 Comparison of performance based on ANN architecture

This section presents comparative results based on varying the number of hidden units in the architecture of all the ANNs in each given ensemble. Ensembles with mixed architectures are not analysed. Again, the Seaton (Cornwall) bathing water is used as a typical example. Ensembles with 5, 8, 12, 18, 27 and 40 hidden units are constructed and ensemble majority decision test results for the 2012 bathing season presented, using SCG training algorithm and the three standard metrics of *AuC*, *F* and E_{opt}.

Figure 5.27 shows areas under the ensemble majority decision ROC curves for the given numbers of hidden units (N_{HU}) in the architectures of all ANNs in each ensemble. Four approaches to assessing the AuC are used:

1. AuC Align ED – based on the threshold alignment technique described in section 5.3.2.2 using the optimum point of Euclidean Distance to select the threshold value for each ensemble member;

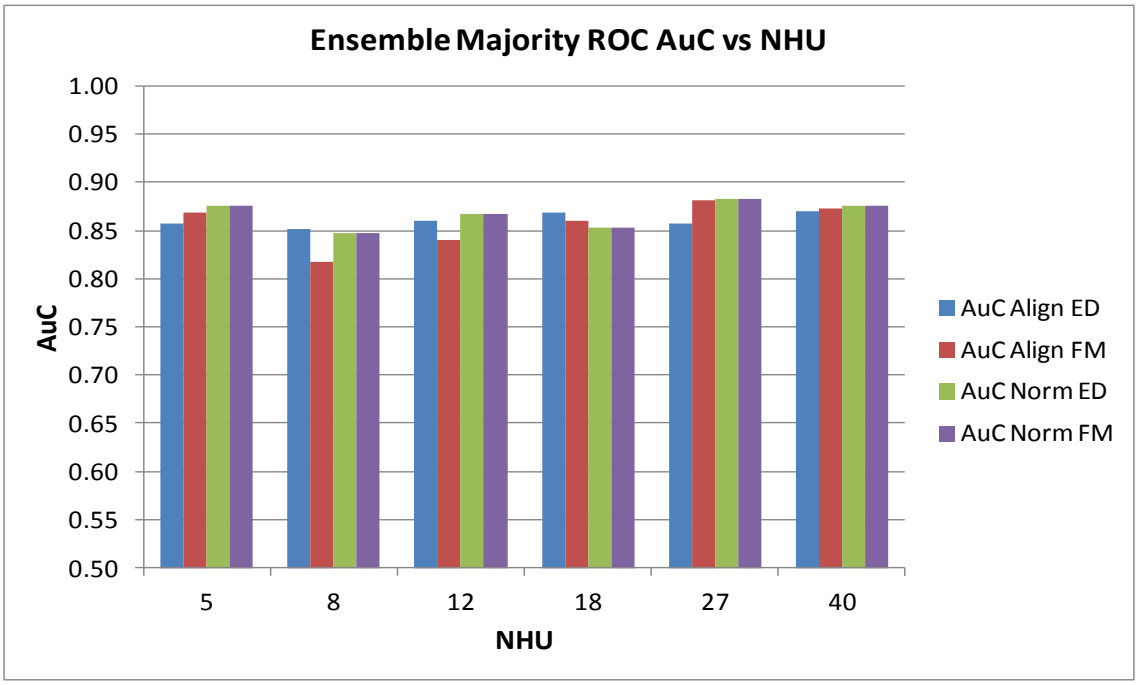


Figure 5.27. Seaton ensemble majority ROC AuC versus numbers of hidden units

2. AuC Align FM – based on the threshold alignment technique described in section 5.3.2.2 using the optimum point of F-measure to select the threshold value for each ensemble member;
3. AuC Norm ED – based on the ANN output span normalisation technique described in section 5.3.2.2 using the optimum point of Euclidean Distance to select the threshold value for each ensemble member;
4. AuC Norm FM – based on the ANN output span normalisation technique described in section 5.3.2.2 using the optimum point of F-measure to select the threshold value for each ensemble member;

This shows an extremely consistent performance as a function of both assessment approach and ANN architecture. N_{HU}=27 produces marginally better performance; hence its use in Figure 5.17 and Figure 5.22.

Figure 5.28 shows results for F-measure, whilst Figure 5.29 presents results for the measure of optimum Euclidean distance. Again, results are reasonably consistent across ANN architectures and approach to assessment. N_{HU}=27 again provides marginally better results in both cases, since E_{opt} requires minimization, whilst F requires maximisation. In the absence of differences of performance between ANN architectures, the rule of parsimony would dictate use of the ANN with the lowest number of hidden units.

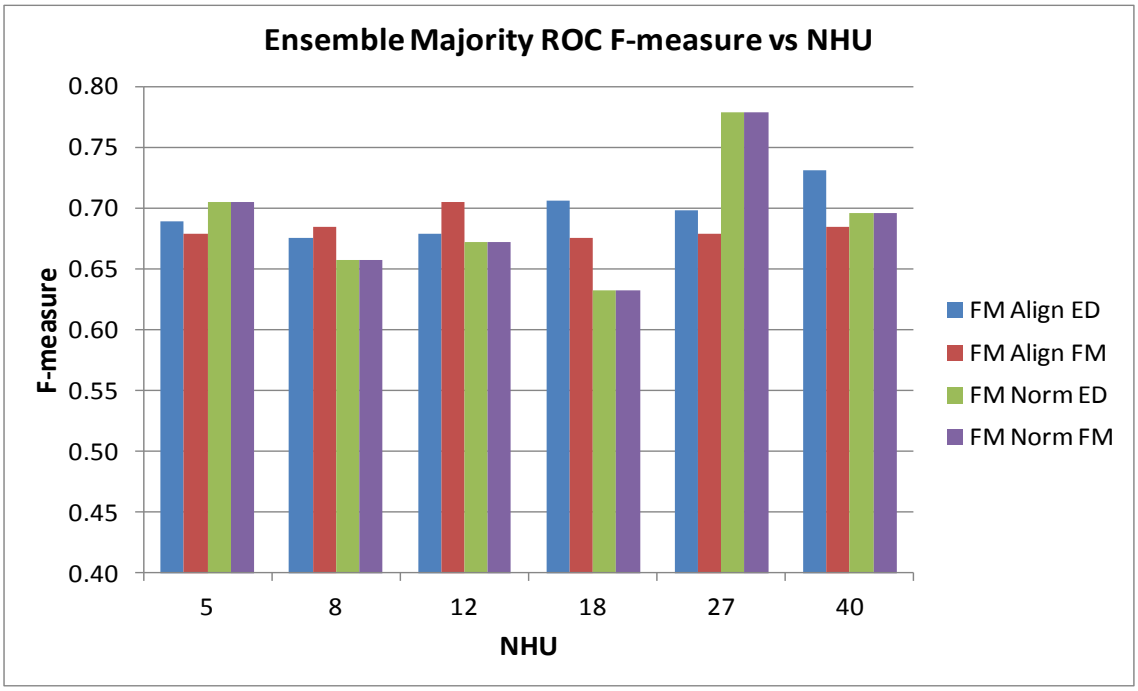


Figure 5.28. Seaton ensemble majority ROC F versus numbers of hidden units

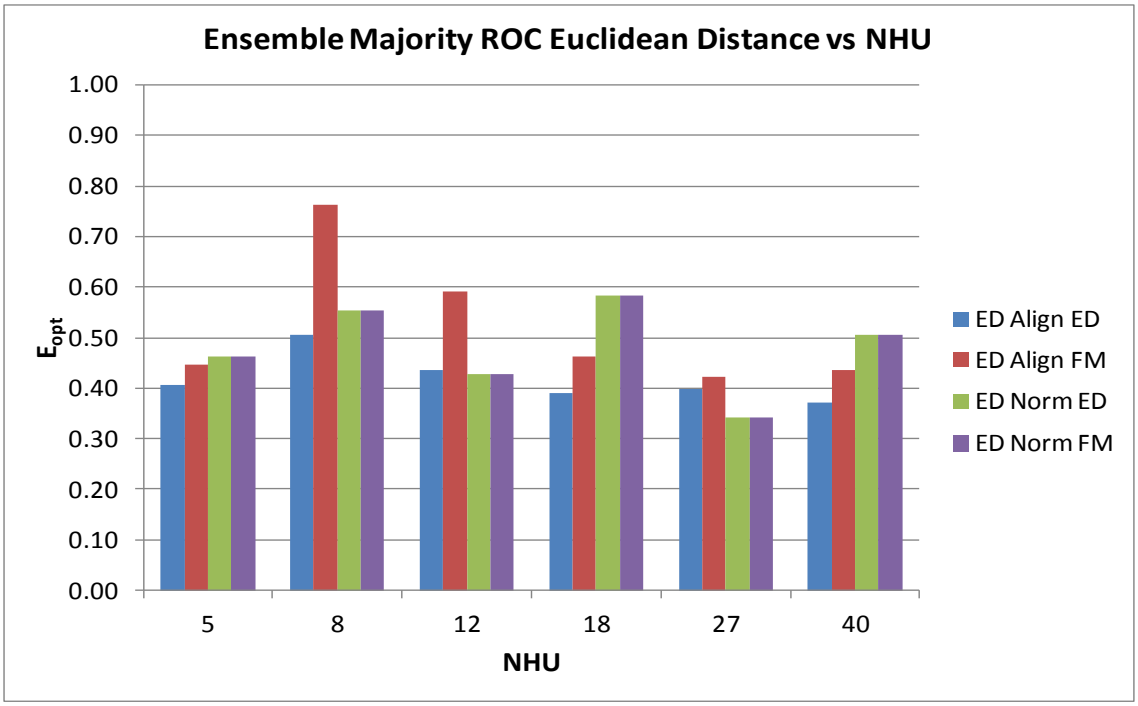


Figure 5.29. Seaton ensemble majority ROC E_{opt} versus numbers of hidden units

The outlier result shown in Figure 5.29 for N_{HU}=8, using the aligned threshold method and F-measure to select the optimum operating point is due to the shape of the ROC in this case, where the lower part of the curve is atypically far to the right (high value of FPR). This results in the value of 'a'=4 in equation 0 playing a significant role in stretching the value of Euclidean distance in this case.

5.3.3 Neural Pathway Strength Feature Selection (NPSFS) Results

This section presents results of trials to determine degrees of relevance of each of up to 12 input features used for the ANN models. This is achieved by analysing the ANN weights and computing combined neural pathway strengths for each input feature. This is described fully in chapter 4 and referred to as CNPSA. By combining these for all members of an NFCV ensemble it is possible to determine degree of relevance of each input feature using the measure EQR (Ensemble interQuartile Range of combined neural pathway strengths). This is applied successfully to the standard UCI machine learning dataset for forest fires in chapter 4. Here, it is applied to the novel Bacti datasets for 5 beaches in south west England provided by the Environment Agency.

5.3.3.1 Seaton (Cornwall) results with SCG / ROC AuC ANN training

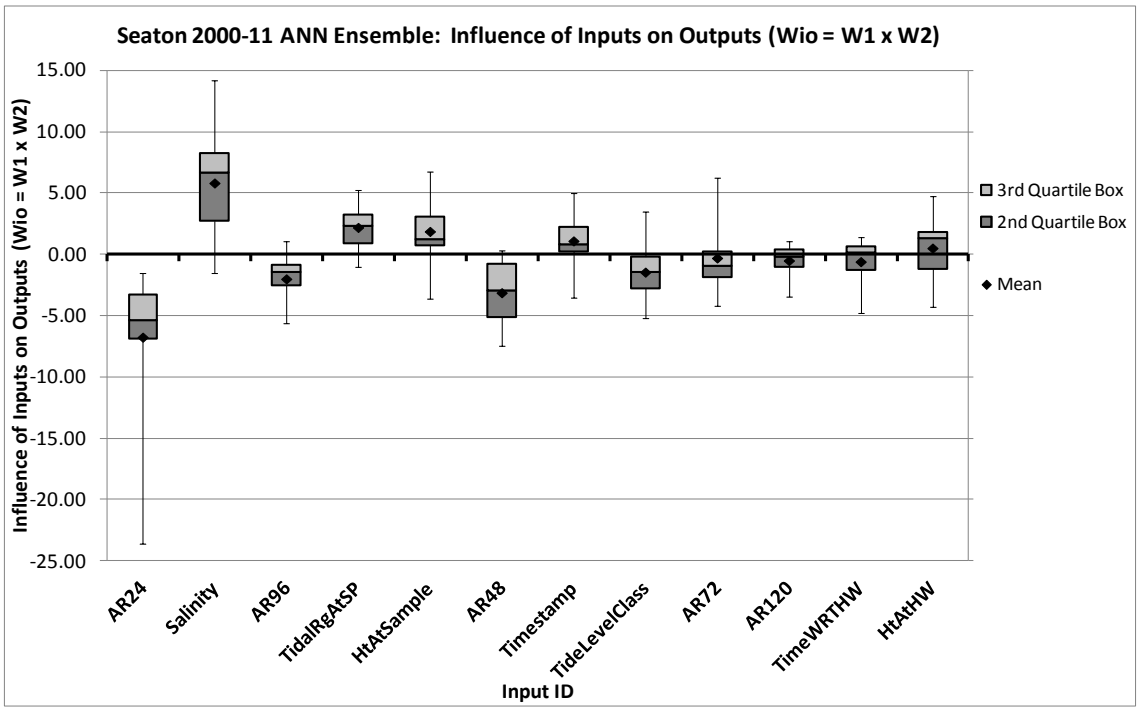


Figure 5.30. Seaton combined neural pathway strengths versus input feature for N_{HU}=5 ensemble

In this trial, the same range of values of numbers of hidden units (N_{HU}) is used: 5, 8, 12, 18, 27 and 40. Figure 5.30 shows the results for the N_{HU}=5 ensemble of 12 ANNs. The measure of EQR is computed for each input based on the bottom of Q2 box and top of Q3 box and their relative positions with respect to the x-axis. The formula is stated in chapter 4. The use of the

interquartile range provides robustness in the technique and immunity from the effects of outlier ANNs. Based on the same results as displayed in Figure 5.30, the EQR values for each input are computed as shown in Table 5.10 and thus the inputs are sorted into ranked order of degree of relevance.

Table 5.10. Seaton: Relevance rank and EQR for 12-input features of ANN with N_{HU}=5

Influence of Inputs on Outputs (Wio = W1 x W2)								
Relevance Rank	**Input Descriptor**	**Mean**	**Max**	**Q3**	**Median**	**Q1**	**Min**	**EQR**
1	**AR24**	-6.761	-1.517	-3.300	-5.356	-6.851	-23.605	**0.482**
2	**Salinity**	5.817	14.196	8.296	6.698	2.777	-1.488	**0.335**
3	**AR96**	-2.005	1.026	-0.833	-1.464	-2.533	-5.658	**0.329**
4	**TidalRgAtSP**	2.193	5.227	3.219	2.325	0.885	-1.050	**0.275**
5	**HtAtSample**	1.866	6.790	3.042	1.257	0.720	-3.592	**0.237**
6	**AR48**	-3.134	0.332	-0.759	-2.915	-5.079	-7.446	**0.149**
7	**Timestamp**	1.092	4.971	2.269	0.857	0.270	-3.572	**0.119**
8	**TideLevelClass**	-1.455	3.521	-0.183	-1.408	-2.771	-5.222	**0.066**
9	**AR72**	-0.298	6.277	0.249	-0.978	-1.901	-4.216	**-0.131**
10	**AR120**	-0.507	1.074	0.384	-0.214	-1.043	-3.438	**-0.368**
11	**TimeWRTHW**	-0.599	1.428	0.674	0.177	-1.282	-4.755	**-0.526**
12	**HtAtHW**	0.507	4.785	1.846	1.285	-1.215	-4.317	**-0.658**

This order is found to vary to some extent with ANN architecture, different randomised initial values of weights and optimisation algorithm. As an example, the results for N_{HU}=27 are now presented in Figure 5.31 and Table 5.11:

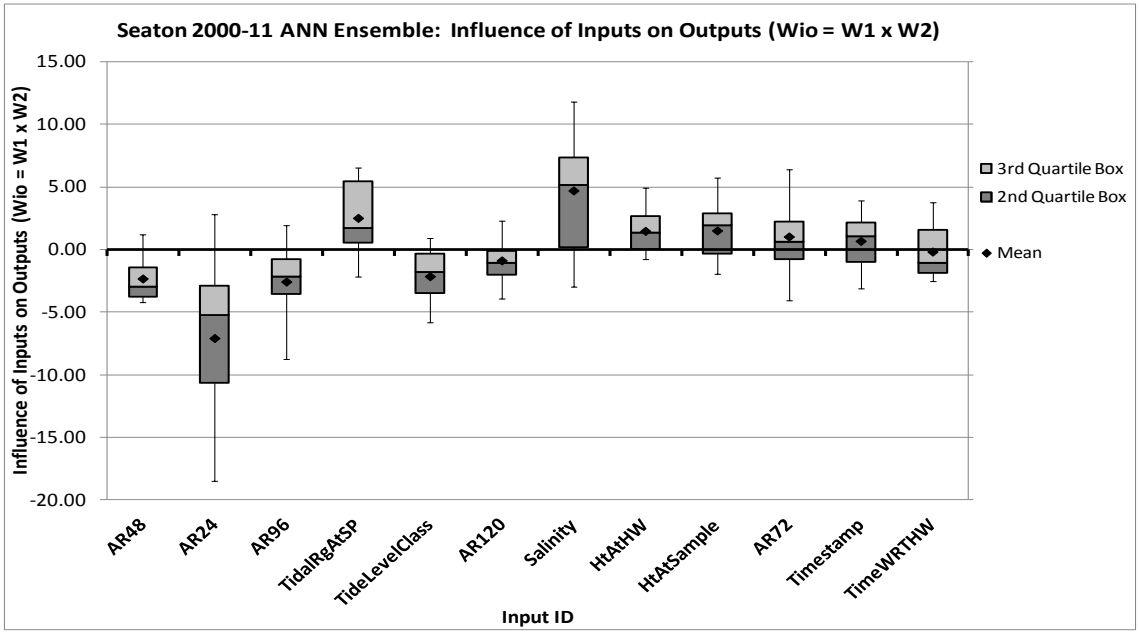


Figure 5.31. Seaton combined neural pathway strengths versus input for N_{HU}=27 ensemble

Table 5.11. Seaton: Relevance rank and EQR for 12-input features of ANN with N_{HU}=27

Influence of Inputs on Outputs (Wio = W1 x W2)								
Relevance Rank	**Input Descriptor**	**Mean**	**Max**	**Q3**	**Median**	**Q1**	**Min**	**EQR**
1	**AR48**	-2.324	1.199	-1.453	-2.945	-3.756	-4.203	**0.387**
2	**AR24**	-7.081	2.816	-2.865	-5.233	-10.674	-18.505	**0.268**
3	**AR96**	-2.570	1.921	-0.754	-2.145	-3.576	-8.781	**0.211**
4	**TidalRgAtSP**	2.516	6.526	5.431	1.716	0.577	-2.170	**0.106**
5	**TideLevelClass**	-2.143	0.892	-0.315	-1.806	-3.448	-5.840	**0.091**
6	**AR120**	-0.881	2.321	-0.135	-1.051	-2.020	-3.941	**0.067**
7	**Salinity**	4.705	11.804	7.345	5.116	0.198	-2.965	**0.027**
8	**HtAtHW**	1.478	4.913	2.651	1.345	0.005	-0.775	**0.002**
9	**HtAtSample**	1.512	5.761	2.901	1.938	-0.350	-1.919	**-0.121**
10	**AR72**	1.028	6.390	2.231	0.630	-0.768	-4.093	**-0.344**
11	**Timestamp**	0.681	3.882	2.125	1.056	-0.971	-3.136	**-0.457**
12	**TimeWRTHW**	-0.172	3.765	1.550	-1.046	-1.898	-2.505	**-0.817**

Despite these differences, a clear pattern of degree of relevance of input features does emerge from analysis of results for a collection of 6 ensembles and in fact only a small spread of degree of relevance occurs for each input feature.

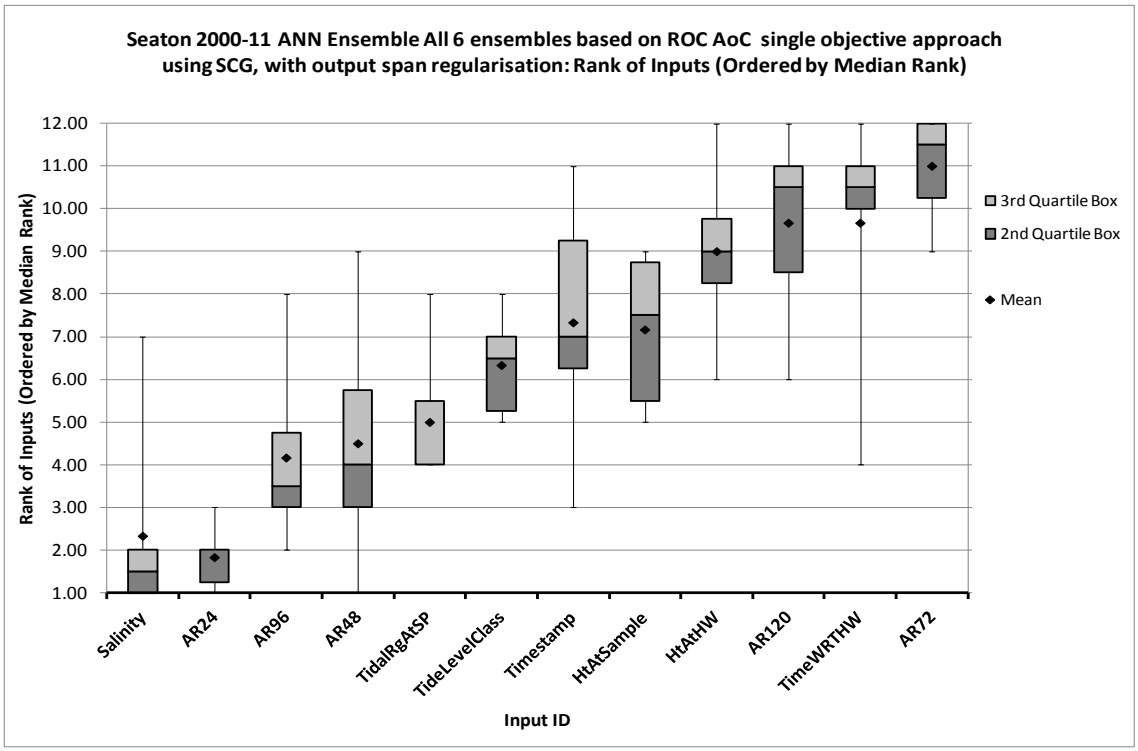


Figure 5.32. Seaton using SCG optimisation: spread of ranks of inputs ordered by median rank

Figure 5.32 illustrates the spread of rank in a box and whisker for each of the 12-input features over the collection of 6 ensembles. These are sorted here in ascending order of median rank.

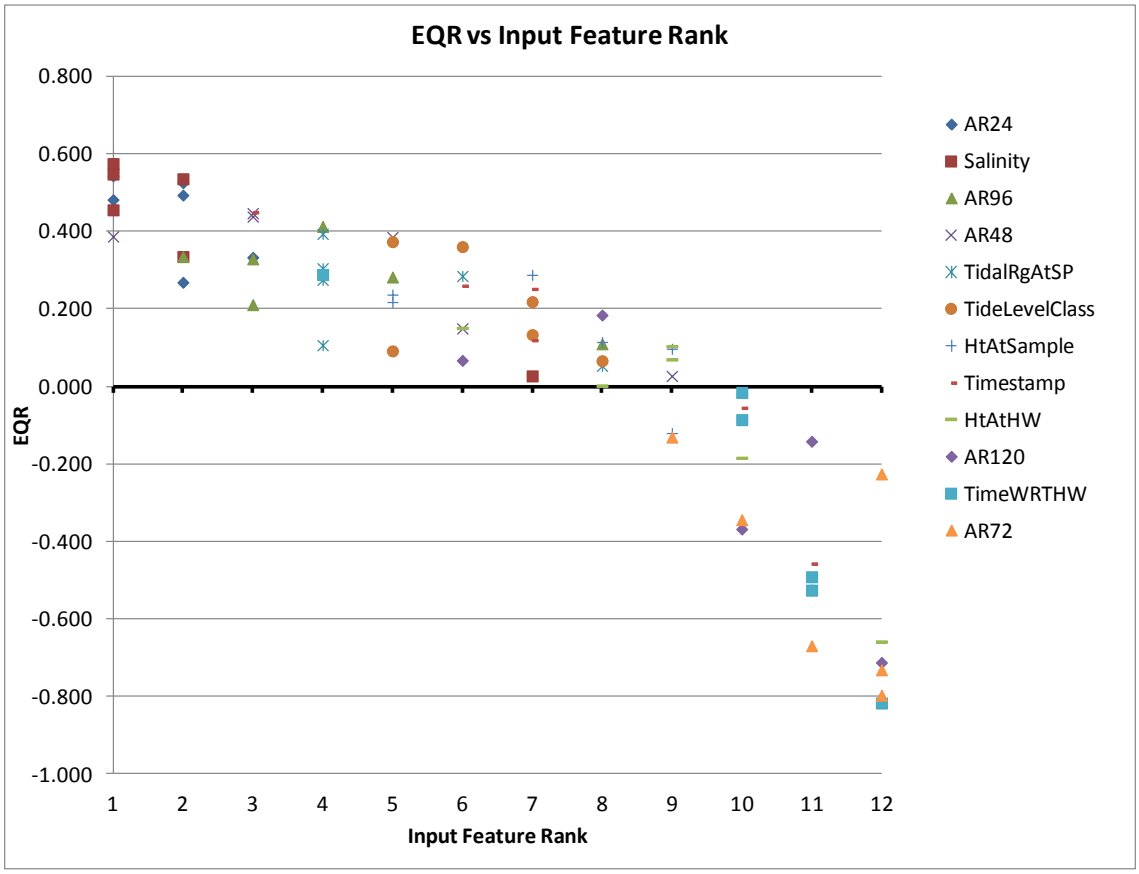


Figure 5.33. Seaton SCG: scattergram of EQR vs. input feature rank for collection of 6 ANN ensembles

Figure 5.33 is a scattergram of EQR versus input feature rank for collection of 6 ANN ensembles of 12 ANNs each. The 12-input features are represented by different shaped markers. The optimisation algorithm is SCG using AuC as described previously. A clear relationship between the rankings and EQRs for each feature over the collection of 6 ensembles is demonstrated. From this raw data, a chart of mean EQR versus mean input feature rank is presented in Figure 5.34 and Table 5.12 and these reveal the underlying structure in the raw ranking and EQR data.

Although this is not necessary for the feature selection method and any single ensemble could be used to rank degree of relevance of input features, the mean rankings presented in Table 5.12 are used to select features to include in or exclude from model ensembles with reducts of the input feature sets.

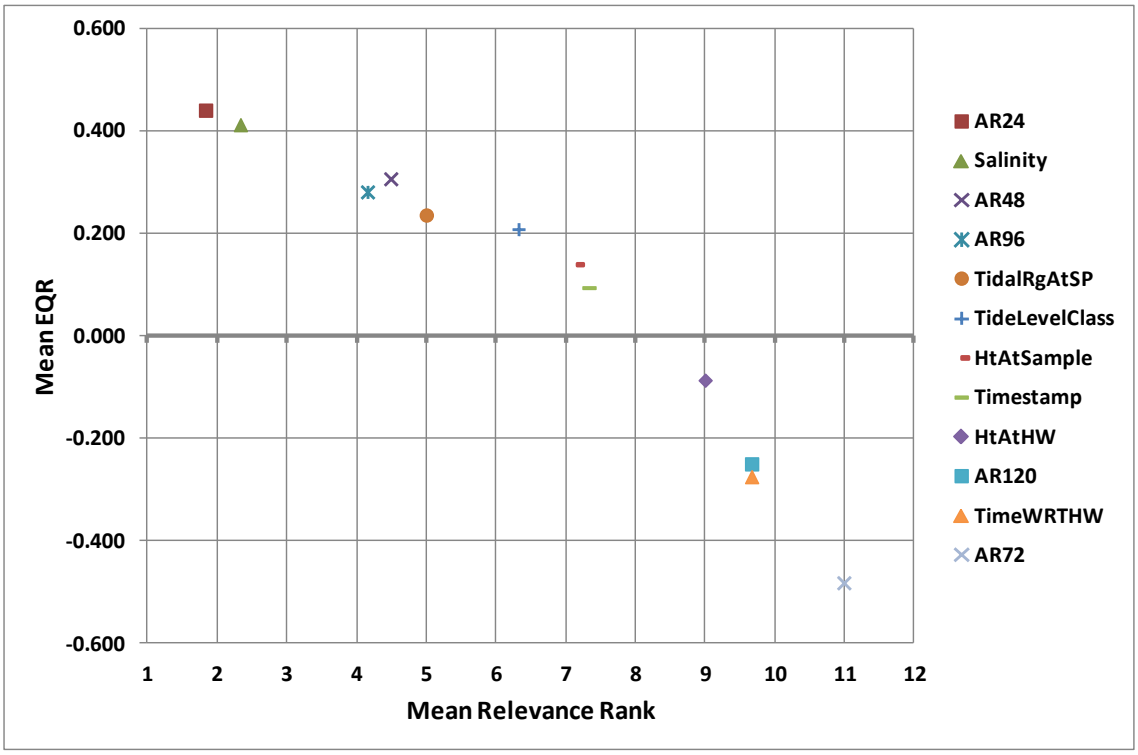


Figure 5.34. Seaton SCG: scattergram of mean EQR vs. mean input feature rank for a collection of 6 ANN ensembles

Table 5.12. Seaton SCG: mean relevance rank and mean EQR for 12-input features

Input Descriptor	Mean Relevance Rank	Mean EQR
AR24	1.8	0.441
Salinity	2.3	0.413
AR48	4.5	0.305
AR96	4.2	0.280
TidalRgAtSP	5.0	0.236
TideLevelClass	6.3	0.207
HtAtSample	7.2	0.139
Timestamp	7.3	0.094
HtAtHW	9.0	-0.086
AR120	9.7	-0.250
TimeWRTHW	9.7	-0.274
AR72	11.0	-0.483

The following secondary trials with the stated reducts of input features are conducted and performance of a collection of 6 ensembles with different ANN architectures is assessed using the standard 3 metrics of ensemble majority decision *AuC*, *F* and E_{opt}. The results are also compared with those of the trial with the full input feature set:

1. The “best” (most relevant) 6 input features (green shading in Table 5.12) – being exactly half the original set of input features
2. The “best” (most relevant) 8 input features (green and amber shading in Table 5.12) – being those with positive values of EQR.
3. The “worst” (least relevant) 6 input features (amber and pink shading in Table 5.12) – being exactly half the original set of input features

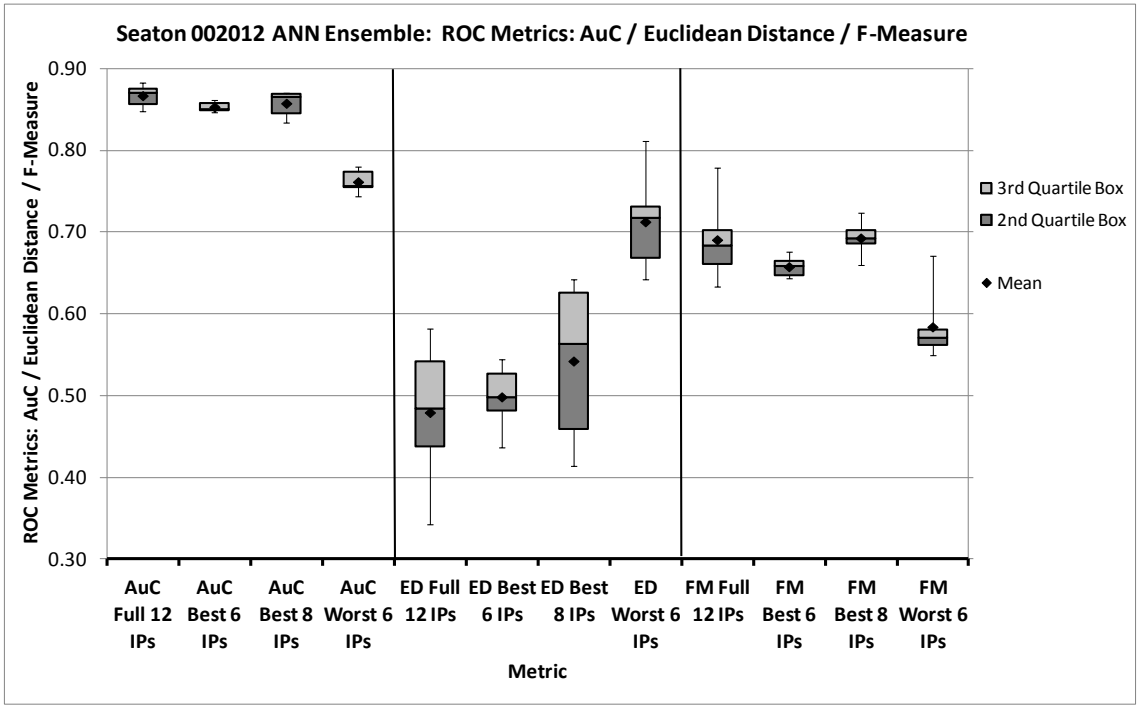


Figure 5.35. Seaton SCG: Performance of full and reduct input feature set ANN ensembles

Table 5.13. Seaton SCG: Comparison of performance of full and reduct input feature set ANN ensembles with 6 different ANN architectures

2-tailed paired T Full-Best 6	0.076				0.595				0.227			
2-tailed paired T Full-Best 8	0.409				0.126				0.938			
2-tailed paired T 6 Best-Worst		0.000				0.000				0.011		
NHU	AuC Full 12 IPs	AuC Best 6 IPs	AuC Best 8 IPs	AuC Worst 6 IPs	ED Full 12 IPs	ED Best 6 IPs	AuC Best 8 IPs	ED Worst 6 IPs	FM Full 12 IPs	FM Best 6 IPs	AuC Best 8 IPs	FM Worst 6 IPs
5	0.876	0.850	0.834	0.754	0.462	0.534	0.642	0.717	0.705	0.667	0.696	0.579
10	0.847	0.851	0.870	0.755	0.553	0.506	0.525	0.735	0.658	0.643	0.704	0.562
15	0.866	0.861	0.868	0.780	0.429	0.488	0.413	0.653	0.672	0.658	0.685	0.671
20	0.853	0.849	0.862	0.757	0.582	0.544	0.602	0.811	0.633	0.676	0.688	0.549
30	0.883	0.846	0.870	0.779	0.342	0.480	0.437	0.642	0.779	0.643	0.659	0.561
40	0.875	0.862	0.840	0.743	0.506	0.437	0.633	0.717	0.696	0.659	0.724	0.581

Figure 5.35 and Table 5.13 document the results from these reduct trials and show the spread of values of AuC (left group of 4) Euclidean distance (centre group of 4) and F-measure (right group of 4) of Figure 5.35. The 2-tailed paired Student’s T-test results at the top of Table 5.13 show that the

populations of results for all 3-metrics are the same (with 95% significance) for the three collections of ensembles with the full input feature set, and the “best” 6 and “best” 8 reduct input feature sets. However, the “worst” 6 input feature reduct models perform worse in the case of each of the three metrics (with >95% significance).

This is encouraging as it suggests that the NPSFS methodology is sound and that EQR is indeed selecting features relevant to the models’ operation and rejecting features that would lead to poorer performance if selected by themselves without the more relevant features. It also demonstrates that more parsimonious models with reduct input feature sets can be constructed in this way without degrading performance significantly.

5.3.3.2 Seaton (Cornwall) NPSFS results with NSGA-II ANN training

The results for the SCG optimisation algorithm used in the previous section are reasonably consistent with those obtained for the NSGA-II based ANN training algorithm. However, the spreads of rankings for each input feature are greater than for SCG using the same collection of 6 ensembles with different ANN architectures. Comparison of Figure 5.32 with Figure 5.36 illustrates this as well as showing the ordering by median rank of input features to be similar but not identical. Similarly, Figure 5.33 may also be compared with Figure 5.37.

These results contribute to the decision to present the performance results using SCG-based ANN training as the main set of comparative results.

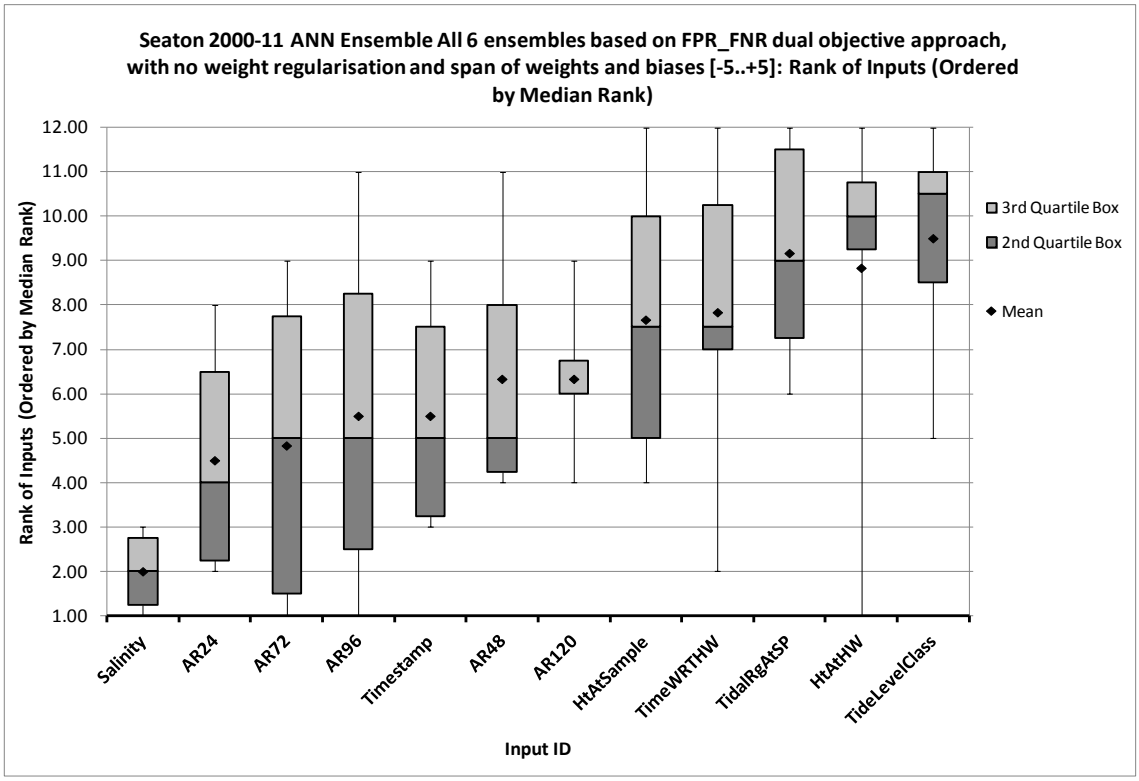


Figure 5.36. Seaton NSGA-II optimisation: spread of ranks of inputs ordered by median rank for 6 ensembles

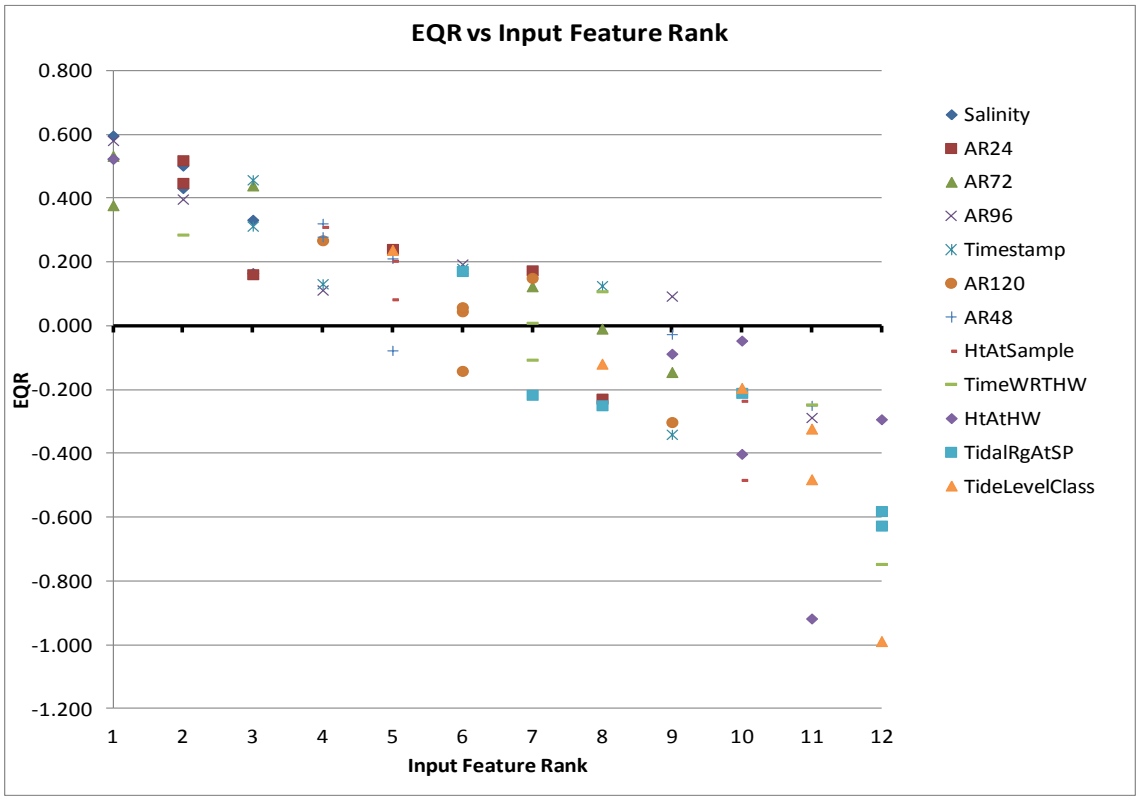


Figure 5.37. Seaton NSGA-II: scattergram of EQR vs. input feature rank for collection of 6 ANN ensembles

5.3.3.3 Neural pathway strength diagram (NPSD) results with SCG ANN training

This section explores the benefits of using Neural Pathway Strength Diagrams (NPSDs) for inspection of the internal operation of the ANNs produced for the experiments already described in this chapter. NPSDs are described fully in chapter 4. However, it is worth recalling that they are scatter grams of weight values for each neural pathway of a 2-layer network; where the x-axis represents weight value for the input of the hidden layer and the y-axis represents the weight value for the input of the output layer. Further, input signal identity is encoded in the shape of the marker, whilst hidden unit number is represented by marker edge colour and output unit number is represented by marker face colour. Also, to recap, there are three breakout views on the data:

- one NPSD per output unit (in this case there is only 1)
- one NPSD per hidden unit
- one NPSD per input feature (signal)

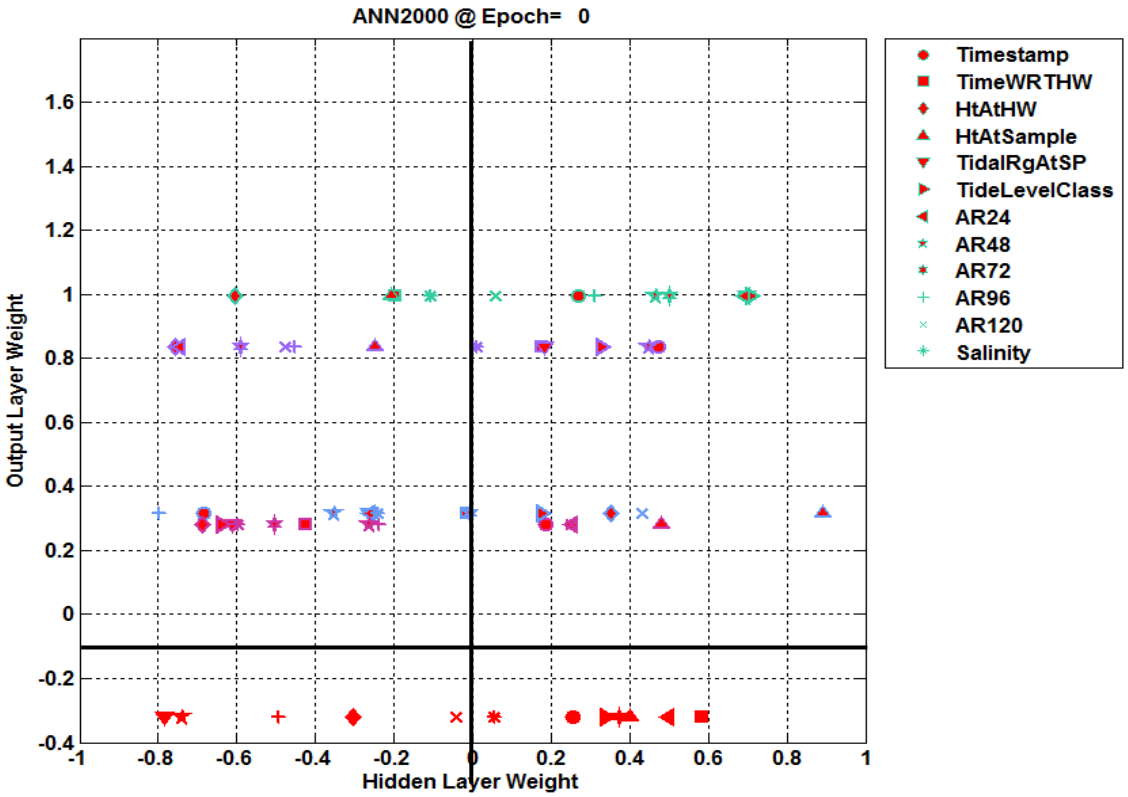


Figure 5.38. NPSD for Seaton ANN2000; NHU=5 prior to SCG training

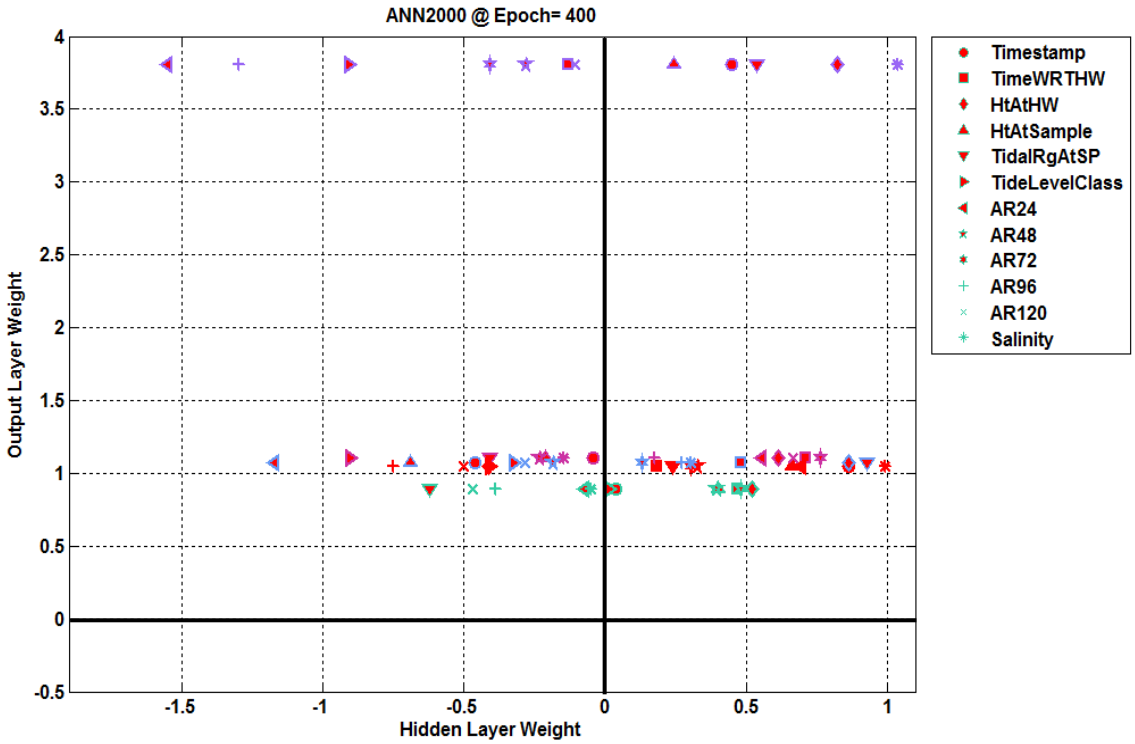


Figure 5.39. NPSD for Seaton ANN2000; NHU=5 on completion of SCG training

Marker colours change in the following spectrum, moving from low index to high for both hidden unit and output unit number: [green → cyan → blue → violet → magenta → red].

Figure 5.38 and Figure 5.39 show the breakout view for the single output node. In both plots, 5 rows of neural pathway markers are clearly visible, corresponding with the 5-hidden units. Each row contains 12 markers corresponding with the 12-input features. It can be seen in Figure 5.38 that all initialised weights have values lying inside the range [-1... +1], whereas in Figure 5.39, following training for 400 epochs, that some weights have moved outside of this range (in both axes / layers) and the output unit's weight from hidden unit 3 (violet marker edge colour) is set to approximately 3.8 (top of chart).

Figure 5.40 is the breakout view by hidden unit and shows the contribution made by each hidden unit to the network as a whole even more clearly. Each hidden unit has its own sub-plot associated with it. All hidden units are contributing significantly to the output, since units 1, 2, 4 and 5 each have a weight from their output to the corresponding input of the single output unit of

around unity. However, the contribution of hidden unit 3 is significantly larger, with the output unit weight of 3.8.

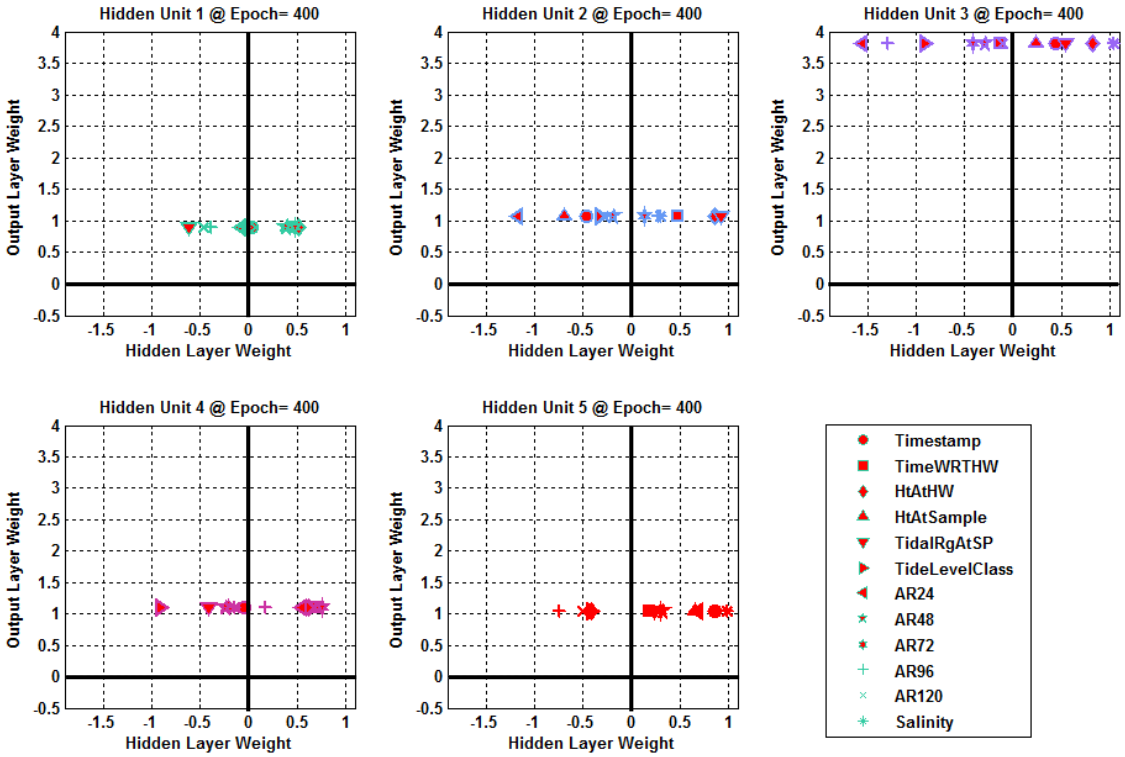


Figure 5.40. NPSD breakout by hidden unit for Seaton ANN2000; NHU=5 on completion of SCG training

Figure 5.42 displays the breakout view by input feature, with a separate sub-plot per input signal. There are 12 of these. Since there are 5 hidden units and a single output unit, there are 5 neural pathways through the ANN from each input to the output. These are represented by the 5 markers on each sub-plot.

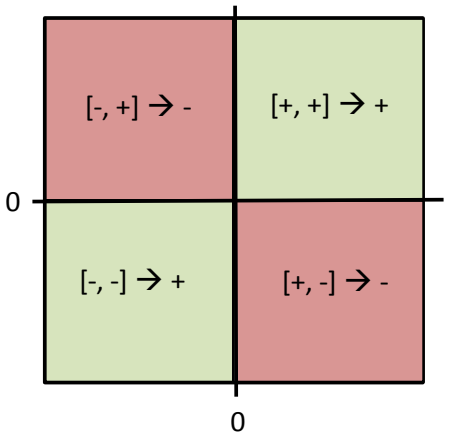


Figure 5.41. NPSD zones of influence

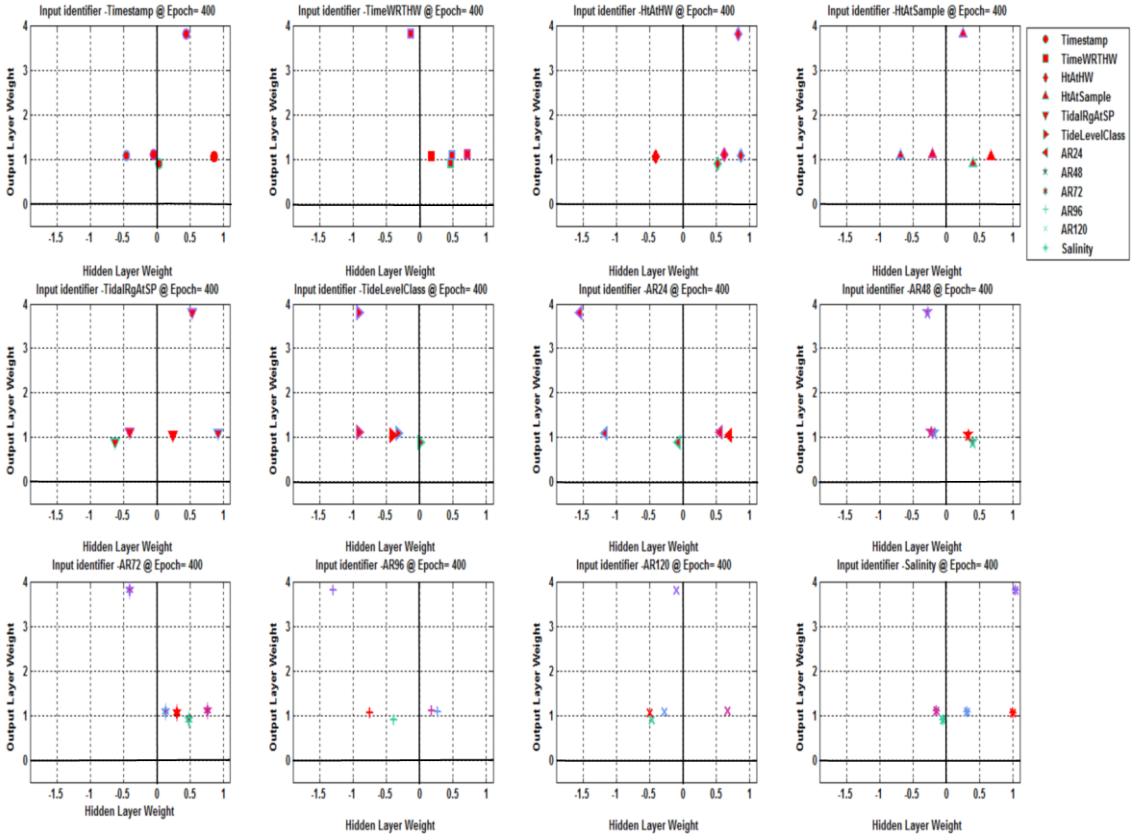


Figure 5.42. NPSD breakout by input feature for Seaton ANN2000; NHU=5 following SCG training

Figure 5.41 illustrates the zones of influence in all the NPSDs. Neural pathway markers in the green areas indicate that those pathways make a net positive (excitatory) contribution to the output, whilst the red areas contain markers that would make a net negative (inhibitory) contribution to the output with respect to the input signal they represent. The example in Figure 5.42 is somewhat asymmetric in that each sub-plot has a y-axis that only extends between -0.5 and +4.0 and none of the markers are in the bottom two quadrants in this case. However, markers to the left of the solid lines (y-axes) represent inhibitory pathways, whilst those to the right of the line represent excitatory pathways. For some input features (e.g. AR120 and AR48) there is a balancing out of some pathway contributions by others in complex conjugate position [54]. Others, such as AR24 and Salinity have a clearer pattern of contribution, with AR24 mainly exerting an inhibitory influence and Salinity an excitatory one. Of course we are only looking here at a single ANN in an ensemble used to produce the EQR input feature relevance values, but

[54] *Markers close to either axis make little contribution as neural pathway strength is a function of $W_1 \times W_2$*

inspection of Figure 5.43 for the ensemble of which this ANN is a member, shows that AR24 and Salinity are the two most relevant input features, whilst AR48 is ranked 6th and AR120 is ranked 10th in relevance.

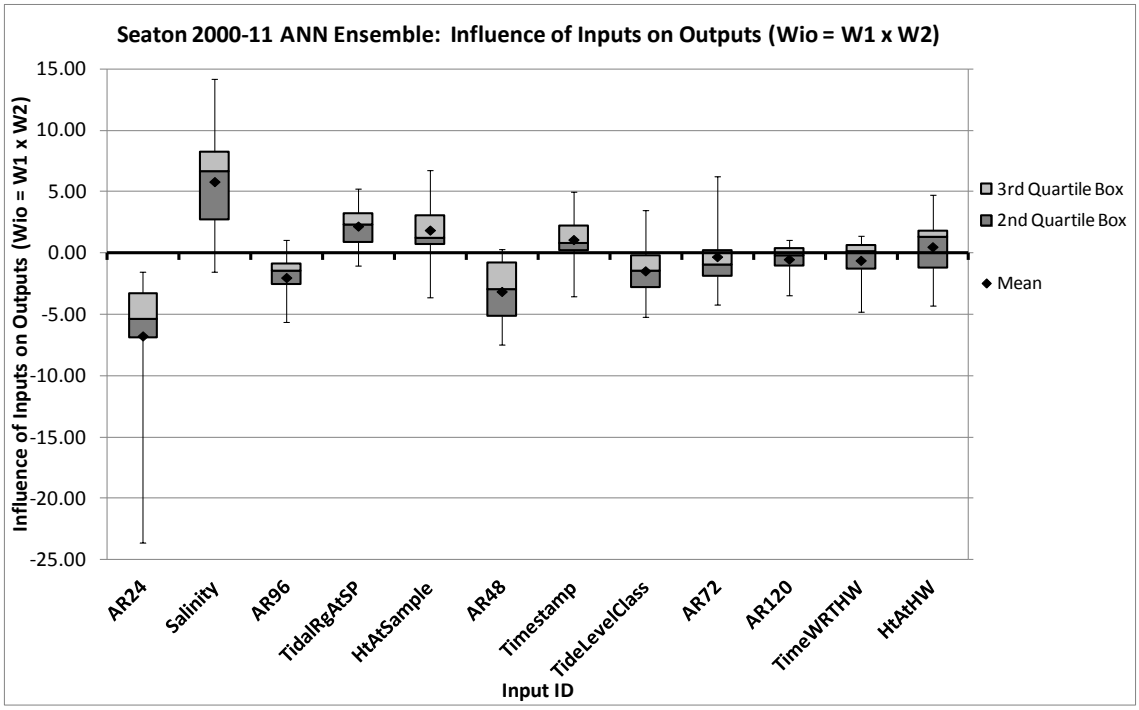


Figure 5.43. Seaton ANN ensemble of N_{HU}=5 influence of inputs on outputs

Therefore inspection particularly of the NPSD breakout view by input feature allows inferences to be made about features likely to emerge as highly relevant on completion of the EQR analysis of the whole ensemble. Figure 5.44 illustrates this for an ensemble member with 27 hidden units. Thus every sub-plot has a cloud of 27 markers representing the neural pathways from the given input to the output via the 27 hidden neurons. The ellipse labelled “A” surrounds the cloud of markers for input AR24 and lies in the inhibitory orientation (ref Figure 5.41). Conversely the ellipse labelled “B” surrounds the cloud of markers for the Salinity input and lies in the excitatory orientation. In complete contrast, the circle labelled “C” surrounds the cloud for AR120, which is not orientated strongly in either direction and is ranked 6th out of 12 inputs for relevance with an EQR of 0.067.

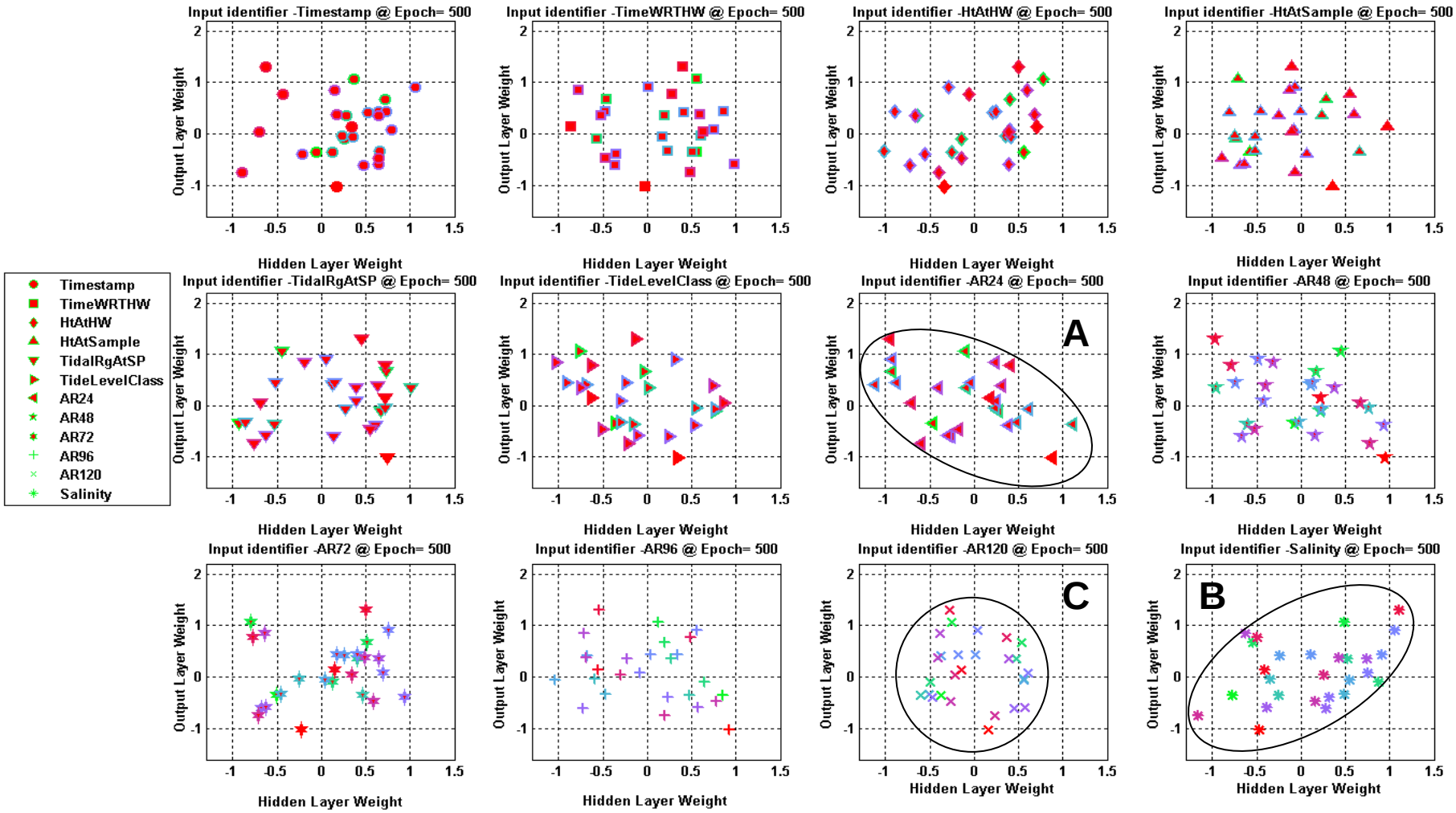


Figure 5.44. NPSD breakout by input feature for Seaton ANN ensemble member with N_{HU}=27

Comparison of NPSDs with Hinton Diagrams

This section compares Neural Pathway Strength Diagrams (NPSDs) with Hinton Diagrams (Rumelhart and McClelland, 1986b) and considers relative advantages and disadvantages of each. It also illustrates some potential benefits of using NPSDs diagnostically.

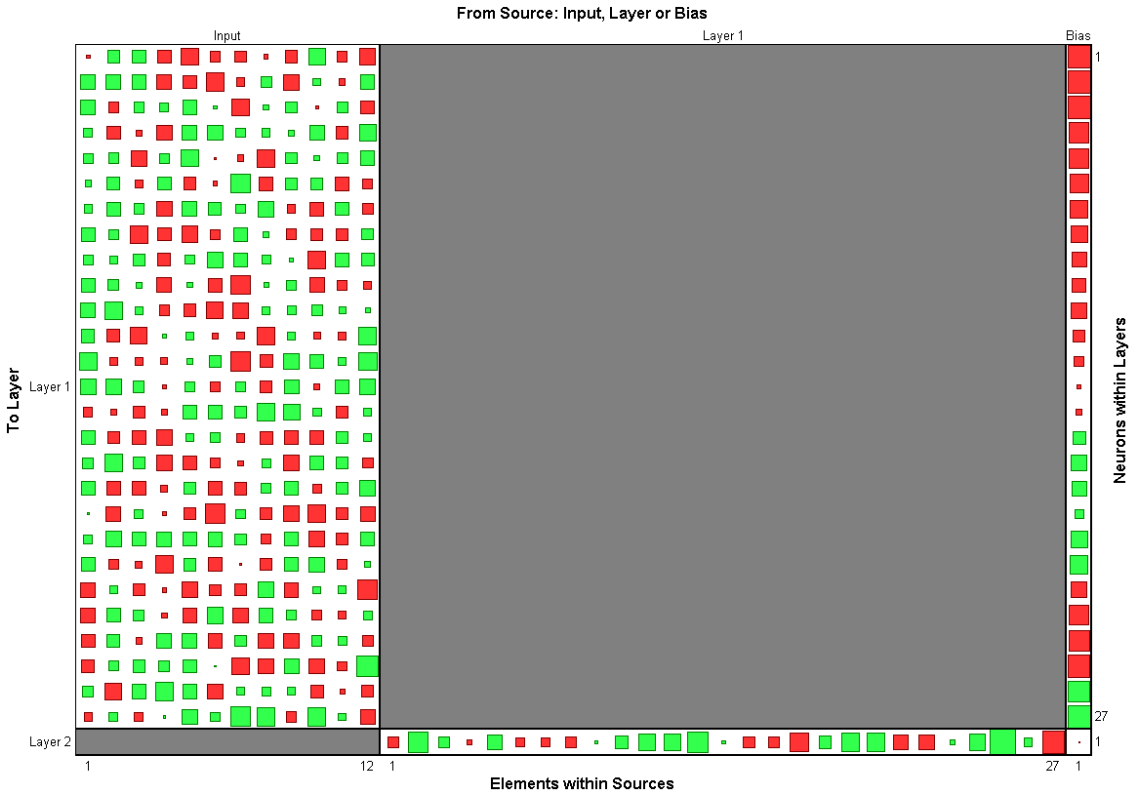


Figure 5.45. Hinton diagram for same Seaton ANN with N_{HU}=27

Figure 5.45 is the Hinton diagram for the same ANN ensemble member as in Figure 5.44. The large rectangle on the left has a matrix of 12 columns (representing the 12 input features) by 27 rows (representing the hidden units). The slim rectangle at the bottom represents the weights for the inputs to the 1-output node from the 27 hidden units. Finally the column at the right represents the bias values for each hidden unit and the output unit.

Squares represent the weight values of each connection in the hidden layer, with size representing the magnitude of weight and colour representing polarity (red=negative; green=positive).

Although information about both weights and biases is well represented, it is not particularly intuitive to work out neural pathway strengths from these

matrices, since the correct hidden layer square must be located and a mental multiplication performed between this and the correct weight in the row vector at the bottom. By contrast, in NPSDs, pathway strengths are represented directly by the position of each marker.

Furthermore, the scaling of sizes of the squares is not absolute, but based on the maximum magnitude of weight present *de facto* in the network at any timestep. This means that comparison of square sizes from Hinton diagram to Hinton diagram is not valid and nothing can be deduced from such a comparison. Conversely, NPSDs represent weights precisely and numerically in the 2 axes of the plot.

However, NPSDs currently ignore biases, whereas this information is included in the Hinton diagrams. Both types of diagram also ignore the effects of the (potentially) non-linear transfer functions on the outputs of the neurons of both layers. Where signals internal to the ANN are driven hard against the extremes of sigmoid activation functions, this can have the effect of vastly increasing weight values in an attempt by the optimiser to compensate. This effect can be observed in NPSDs, as the scales of the axes would indicate the high weight values. However, in Hinton diagrams it may appear that most weight values are small; but this is only relative to perhaps one or a few weights that have attained very large values.

Figure 5.46 illustrates an NPSD breakout chart by input feature for a single outlier ANN from an ensemble of 7 ANNs with N_{HU}=27 for Readymoney beach. It can be seen that there are 8 input features in this trial; one per sub-plot. Weight values of up to +/-5 in both axes lead to very high magnitudes of combined neural pathway strengths of >|100|, especially where a large number of (in this case 27) neural pathways combine to connect each input signal to the output. These pathway strengths are summed. Table 5.14 and Figure 5.47 detail these combined neural pathway strengths and show that for all 5 antecedent rainfall (*ARn*) inputs, pathway strengths are atypically high.

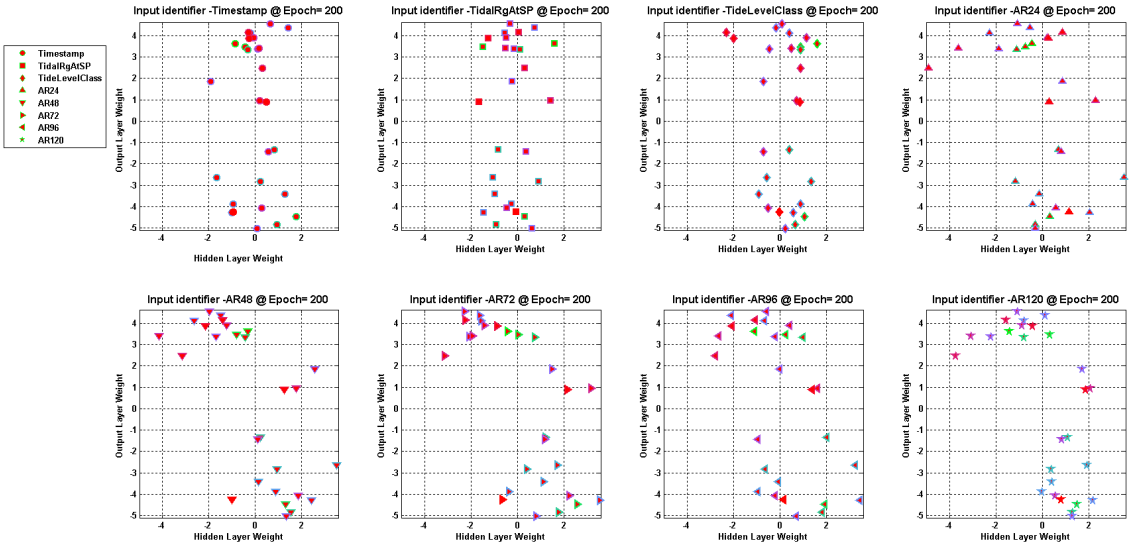


Figure 5.46. NPSD for Readymoney ANN2005 outlier with very high neural pathway strengths

Table 5.14. Readymoney: Combined neural pathway strengths by input feature for outlier ANN

Input Feature	W1 x W2
Timestamp	-5.86
TidalRgAtSP	5.54
TideLevelClass	-8.42
AR24	-66.06
AR48	-121.00
AR72	-111.58
AR96	-78.17
AR120	-90.77

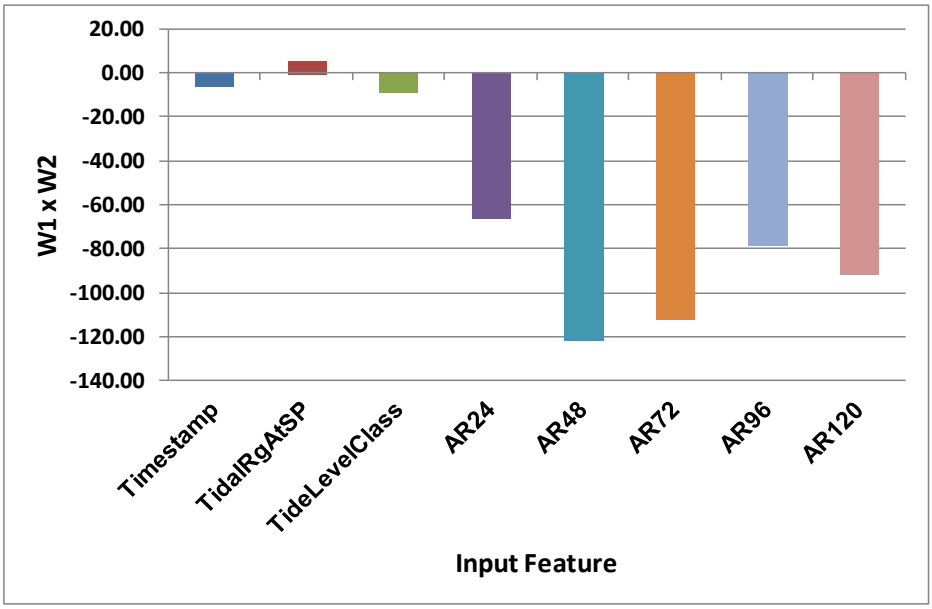


Figure 5.47. Readymoney: Combined neural pathway strengths by input feature for outlier ANN

This shows up in the box and whisker plot for this ensemble, used to calculate EQR for each input feature (Figure 5.48). The outlier ANN is revealed

in the long whiskers on the 5 rainfall inputs. However, it can also be noted that, despite there being only 7 ANNs in this ensemble, the method of calculating EQR using first and third quartile boxes makes the EQR measure tolerant of such an outlier in the ensemble:

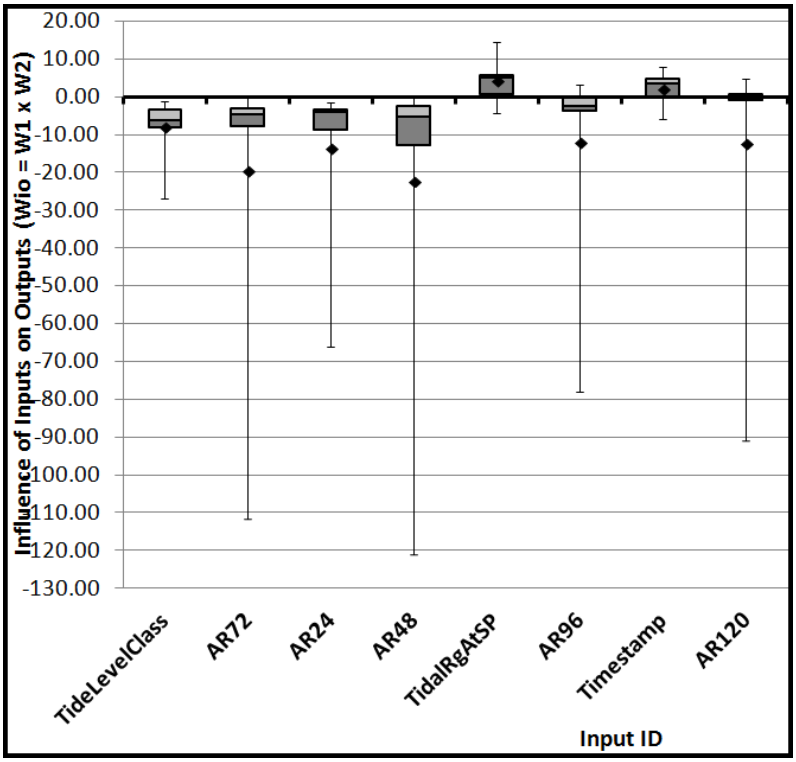


Figure 5.48. Readymoney: Combined neural pathway strengths versus input for NHU=27 ensemble

By contrast, the Hinton diagram (Figure 5.49) for the same ANN does not give direct indication of these high pathway strength values. Instead, the high relative values of weights in the output neuron (shown in the row at the bottom) have to be taken in combination with the high relative values of weights present in the 5 rightmost columns of the hidden layer weights matrix at the left. However, because of the normalised size scaling of the squares, there is no way of estimating absolute levels of pathway strength from this.

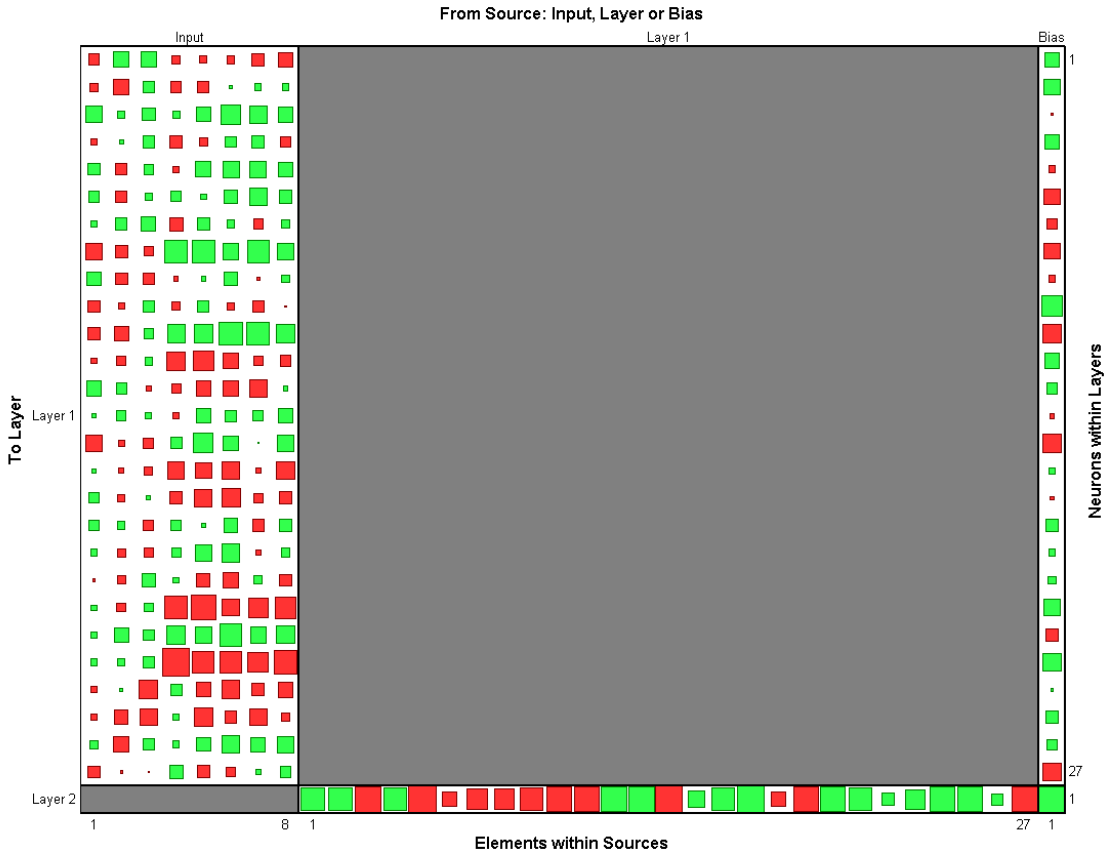


Figure 5.49. Readymoney: Hinton diagram for outlier ANN

Finally, it is perhaps worth observing that for an ANN such as this one, over half the area of the Hinton diagram, in the form of the grey square, contains no information whatsoever. This area could be used to represent neural pathway strengths as it is a more than adequate size for doing so. Another matrix 27 x 8 would be needed in this example as there are 27 terms that are summed to form each of the 8 rows in the combined neural pathway strength matrix shown in Table 5.14. However, for multiple output ANNs multiple additional matrices would be needed. However, this would still suffer from the normalised relative scaling problem.

5.4 Discussion and Further Results

ANN ensemble construction with evolutionary algorithms

As an alternative to the NFCV approach to ensemble generation used throughout, the use of NSGA-II or other evolutionary algorithms might potentially allow the possibility for a range of non-dominated solutions from a single population to be selected as ANN ensemble members on completion of the training. The question arises as to why this has not been done, since the population of ANNs has been generated anyway.

This may seem to have advantages in that it would be more computationally efficient than creating a whole population of (usually at least 100) solutions for each ensemble member and then rejecting all but the single "best" solution for inclusion in the ensemble. However, for each NFCV data-fold, all EA population members are trained using the same data, so differences in final weight and bias values between population members can only be attributed to different (randomised) starting points in the decision space resulting in different population members exploring different regions of the decision space. As there are no differences in the input data, this is unlikely to reveal pattern in the differences in the ways that these ensemble members have treated each input feature. The hypothesis is that all input features will tend to be treated similarly and so be found all to be relevant, using EQR as the measure.

The automated Neural Pathway Strength Feature Selection (NPSFS) methodology, on the other hand, relies on the ensemble members having been trained on different, if overlapping, subsets of the overall training set This allows the Combined Neural Pathway Strength Analysis (CNPSA) to be evaluated across the ensemble of ANNs to identify (through use of the EQR metric) those inputs used similarly by the majority[55] of ANN ensemble members. That is also to say for the majority of training data subsets.

[55] *For the case of EQR>=0, this means 75% or more of ensemble members treat the given input in the same sense (excitatory or inhibitory)*

The approach adopted is to select a single “best” ANN from each population for inclusion in the NFCV ensemble, when using NSGA-II. This makes the methodology independent of the optimisation algorithm used for training and single-threaded GD-based approaches work equally well.

In order to test the possibility of using NSGA-II populations as NPSFS ensembles, the CNPSA / EQR methodology is applied to a collection of 12 populations (treated as ensembles) of unique ANN solutions on completion of NSGA-II training is carried out for the Porthluney catchment. Each population is trained on a single training data-fold (for a given test year 2000-2011). Each population has on average 8.33 unique members (range 5 to 14) on completion of training.

The following summary results for EQR versus input feature relevance rank are obtained as shown in Figure 5.50:

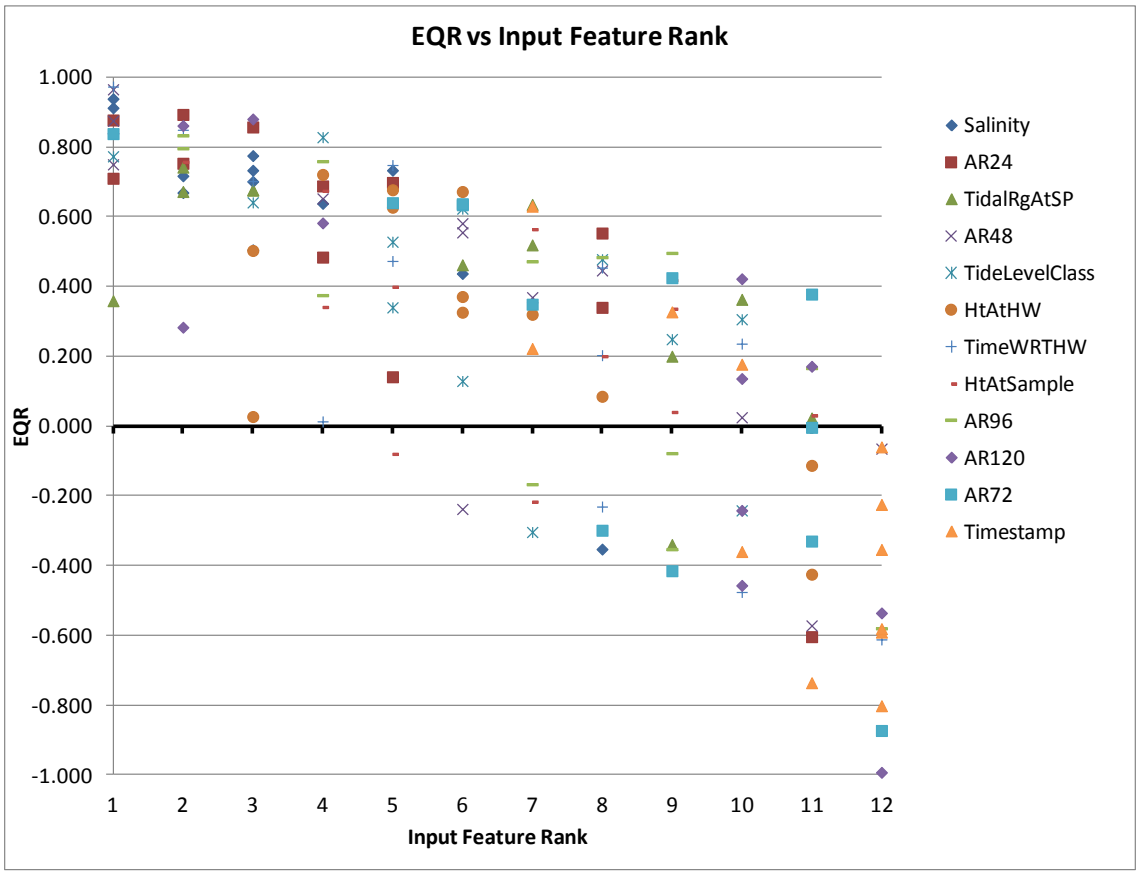


Figure 5.50. Porthluney: NSGA-II populations EQR versus input feature relevance rank

Finally, the mean relevance rank for each input feature over the collection of populations is computed and displayed in Figure 5.51:

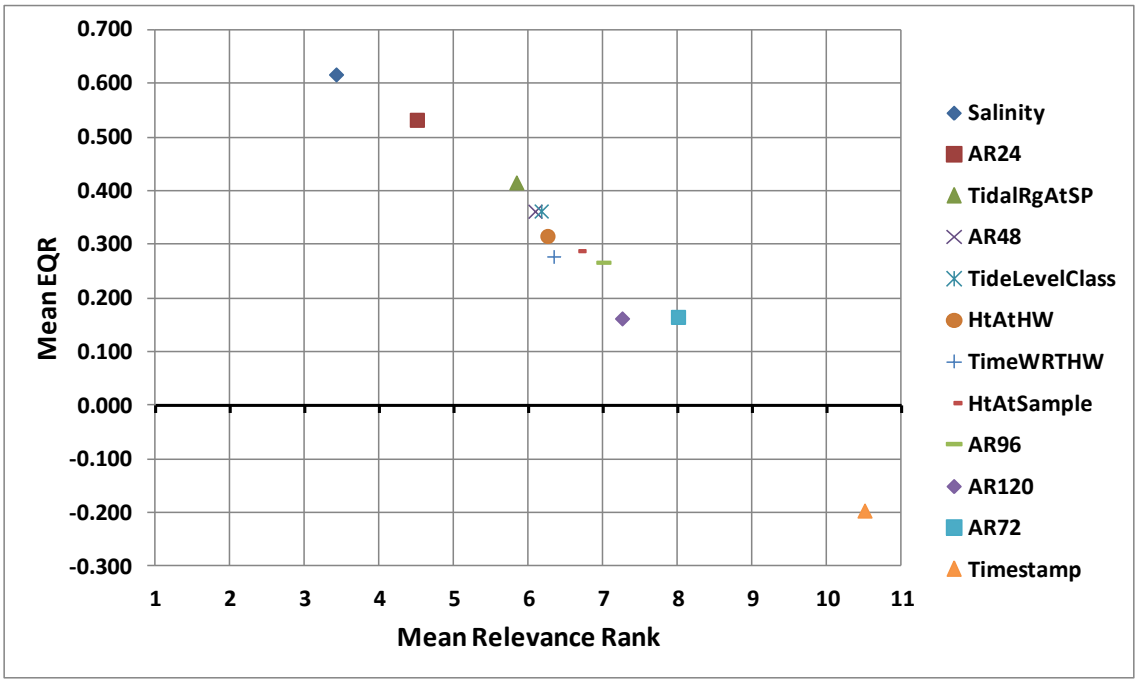


Figure 5.51. Porthluney: NSGA-II populations mean EQR versus mean input feature relevance rank

Figure 5.50 shows that there is a huge spread of both input relevance rank and EQR values and that EQR is in many cases holding positive even when the rank is as high as 11. It is useful to contrast this with Figure 5.33 or Figure 5.37, which, even though for a different beach, show typical performance for NFCV ensembles. Inspection of Figure 5.51 reveals that all but one input feature (Timestamp) do indeed have positive values of EQR when treating the NSGA-II populations of ANNs as potential ensembles. Mean ranks of input features are also much closer clustered to the centre of the range of relevance rank.

These factors combine to demonstrate that use of EA populations as ensembles for feature-selection would be inferior to using the NFCV ensembles trained on overlapping but different subsets of the Bacti dataset available for a beach. The implication is that the CNPSA / EQR method is working as well as it does, because of the use of different datasets in the training of each ensemble member.

Comparison of DT, simple threshold and ANN ensemble models

This chapter presents a range of comparisons between performances of the three types of model for the five designated bathing beaches for which daily test data for the 2012 bathing season are available.

In the case of DT and simple threshold models, there is just a single operating point and this has been optimised by work between the Environment Agency and other stakeholders involved in the bathing beaches. The optimisation has to some extent been carried out to keep the potential number of bathing advisory notices issued in a bathing season by beach managers to a level acceptable to them. As a result, the operating points of these models can be seen to favour false positives rather more than false negatives. On the ROC curves they are all positioned very much at the top and to the right.

The approach taken in this study is either:

- to minimise both FPR and FNR simultaneously (in the case of NSGA-II), taking into account the value of a=4 recommended in the SEPA study (Stidson et al., 2012); the relative importance of false positives to false negatives; or
- to maximise ROC AuC (in the case of SCG optimisation algorithm) and then find optimum operating point on the ROC using either F or E_{opt} again taking into account the value of a=4.

This has found significantly different optimum operating points for the ANN ensemble majority ROC-based models than for the DT and simple threshold models. As a result, ANN performance using F or E_{opt} as metrics is shown to be consistently better than the *de facto* DT and simple threshold models. Nonetheless, the DT and simple threshold models sometimes perform above the ANN majority ROC curves. It is not known whether changing the DT operating points to nearer to the ROC optimum would result in better or worse performance than the ANNs, whereas (at least for Porthluney beach, where an ROC has been constructed for the simple threshold (Figure 5.13 and Figure 5.14)) simple threshold performance would be similar to the ANNs with the currently available input datasets and models.

5.5 Conclusions and Future Work

This Bacti case study provides an excellent benchmark allowing demonstration of the NFCV / NPSFS approach to selecting input feature reducts based on their degree of relevance, using the EQR metric. The reducted models perform either equally well or sometimes better than the original models with the full input set, but enjoy the benefit of being simpler, more parsimonious models.

The ROC scenarios used in the trials demonstrate the ability to find the optimum operating point (threshold) for each model. The task of combining models in an ensemble is evaluated using two alternative approaches: normalisation and alignment, of which the normalising method appears to be the more robust.

Two approaches to training the ANNs are demonstrated:

- using NSGA-II evolutionary algorithms with dual objectives of FNR and FPR as costs to be minimised
- using SCG algorithm with ROC AuC and output span regularisation as a combined single objective

Both of these are effective, but marginally better results are produced overall using the SCG methodology trialled here. Occasionally NSGA-II performs better, but the conditions under which it does this are not yet clear; so there is potential for further research on this.

Despite the better performance of the ANN models than the DT and simple threshold models, given their current operating points, the trials are overall inconclusive about the relative performances of the three models types, given the possibility for changing the operating points of the DT and simple threshold models. Nonetheless, it is possible to consider using additional or different input features to improve performance of the ANN models further. A robust method of testing any new input feature for relevance is now demonstrably available; so this is seen as a major contribution of this case study and thesis.

There is considerable potential for future work to explore a number of additional input features, including perhaps, meteorological features, such as octal cloud cover, wind-speed, direction, air temperature, atmospheric pressure, relative humidity, UV levels and predictions of rainfall. Additionally, hydrological / oceanographic parameters could be potentially included, such as wave height, turbidity, sea surface temperature, dissolved oxygen, river flow rates, CSO spill data and salinity of streams (where present).

The models created to date are for individual beaches, but there is considerable merit to attempting to construct and assess combined models that cover more than one or even many beaches. These may possibly need to employ time-invariant catchment characteristic data as inputs to the models. These could include such factors as catchment total area, percentages under various land uses (including various rural and urban), catchment average steepness, lengths of watercourses, numbers of / distances to CSOs from the beach. Some of this data is already available from "source apportionment" project work already undertaken by / through the Environment Agency.

One of the main benefits of attempting the construction of such models would be the potential to simplify future model development to work towards providing high quality models for the 608 designated coastal bathing beaches in the UK. Further improvement of model performance may also result.

Finally, the work conducted in this case study used multi-layer perceptrons (MLPs) with 2-layers of neurons. Use of other types of machine learning models could also be explored. These could include deeper MLP networks, Bayesian Belief Networks (BBNs), Support Vector Machines (SVMs) and/or Relevance Vector Machines.

Chapter 6: Conclusions

This chapter summarises and discusses the novel methodologies described in this thesis. It is organised as follows:

The *Conclusions on novelty claims* section refers back to the claims of novelty for this thesis in the introductory chapter and makes observations on these. The *Discussion* section comments on and qualifies these claims; whilst the final section on *Future work* proposes a number of possible opportunities for research leading on from the contributions covered in this thesis.

6.1 Discussion on claims of novelty

It is hoped that the novel techniques outlined in this thesis ultimately make a contribution to the challenge of the widespread adoption of machine learning-based modelling techniques such as ANNs for live, real-time systems. By opening up the black box and revealing useful structure in the information learnt by the ANNs during training, it aims to encourage a move beyond "grey box modelling" into what might be termed "rainbow box modelling". There is no reason why the parametric output of automated learning algorithms needs to remain hidden from researchers in a "black box". Neither are the "hidden layer(s)" of a neural network really hidden. Their weights and biases are as available for inspection as are those in the output layer. Other researchers have also studied hidden layer unit outputs (Jain et al., 2004; Sudheer and Jain, 2004). Notwithstanding the specific validity of the black-box approach to the analysis of network transfer functions; in retrospect perhaps the ubiquitous use of the above terminology within the machine learning community has not always been very helpful? This is especially in respect of efforts trying to encourage the uptake of ANNs as a data driven modelling approach to problems in hydrology and the environment.

The claims of novelty made in this thesis are now discussed and qualified in a little more detail as follows:

6.1.1 CNPSA

"Combined Neural Pathway Strength Analysis" (CNPSA) combines the effects of the weights in 1HL feedforward ANNs by summing the strengths of all possible neural pathways from each input to each output. This is achieved simply and computationally efficiently by a single matrix multiplication. This approach neglects the effects of the non-linear activation functions on the outputs of each neuron, but this is far from unprecedented. Other approaches have also neglected these, such as Hinton diagrams (Hinton, 1984; Hinton et al., 1993) and the methodology proposed by Olden and Jackson (2002).

Despite this, CNPSA has been demonstrated as valid in terms of its ability to reveal structure in the input-to-output connectivity within 1HL feedforward ANNs. It distinguishes the overall sense of use (inhibitory / excitatory) of each input as well as quantifying the magnitude of its overall influence. By contrast with Hinton diagrams, the emphasis is on neural pathways through the network, rather than individual weights; so it arguably provides additional insight into the structure of any given ANN, not provided by Hinton diagrams. It also incorporates a quantitative measure of influence of each input, which Hinton diagrams do not, in an absolute sense, since they automatically normalise the sizes of displayed weight boxes based on the range of weights found throughout the network.

As a diagnostic tool, CNPSA has also been demonstrated to be useful for identifying outlier ANNs in ensembles of models as well as problems down to the level of individual neurons and connections within an ANN.

As currently implemented, the values of biases for the neurons are neglected in CNPSA. Despite this, the method has been demonstrated as effective. Biases can be regarded as special cases of weights, connected to an extra fixed value input of 1, so could readily be included. This has not been done, since the biases are not candidates for feature selection or pruning.

6.1.2 NFCV / EQR

N-fold cross-validation (NFCV) is a well-tested technique for ensemble creation as well as a thorough approach to model performance evaluation.

Ensemble creation using this approach relies on statistical differences between training datasets in order to produce variety in the set of models comprising the ensemble. If, despite this variety, the (inhibitory or excitatory) sense in which a given input is used is consistent across a large majority of ensemble members, this can arguably be classified as "a relevant input". The converse is also likely to be true. This intuition is used as a basis for the novel Ensemble interQuartile Range (EQR) metric for measuring relevance of input features based on variability of CNPSA neural pathway strengths across all the members of the ensemble. EQR has been demonstrated to permit reasonably consistent ranking of input features by relevance and, despite variability in these rankings from ensemble to ensemble, there is arguably sufficient consistency to allow the feature-selection methodology below to work with the results of a single ensemble. In practice, in the experiments conducted here, a mean ranking of a collection of between 6 and 15 ensembles has been used to good effect. Even at this level, the method remains far more computationally efficient than that proposed by Olden and Jackson (2002), for example, and compares favourably with techniques based on bagging and/or boosting or extraneous measures such as MI, PMI or R^2 correlation. These methods still require models to be built and calibrated in addition to the computations they involve.

6.1.3 NPSFS

The use of CNPSA and EQR to rank inputs allows two selection strategies to be employed: a) select inputs with EQR>0; b) select the highest EQR-ranked n inputs. This is referred to here as “Neural Pathway Strength Feature Selection” (NPSFS). This has successfully been used to select relevant inputs and construct a number of more parsimonious models with reduct input feature sets. These have been demonstrated to perform at least as well as the original models with the full input-feature sets; or in some cases better. A control experiment has been conducted in which a reduct of "least relevant" (selected by EQR<0) input features has been used as inputs for an ensemble of models. This has demonstrated markedly inferior performance across the ensemble – indicating that EQR is a valid approach to ranking relevance of input features.

NPSFS is used in two examples in this thesis: in chapter 4 it is demonstrated with regression models predicting the area of forest fires from a

UCI dataset. In chapter 5 it is employed with classifier models predicting bathing water quality at beaches. In both cases, the ANNs used have a single output to predict a single quantity or class label. Therefore the matrix $W_{io} = W_1W_2$ (product of the ANN layer weight matrices) is a column vector. This makes the EQR a scalar value for each input. In the case of multi-output ANNs, the EQR value would be a row vector for each input, with one element per ANN output. A further strategy for analysis of the EQR vectors would have to be developed, so that inputs could be selected based on relevance across outputs. This has not been attempted in this thesis.

6.1.4 NPSD

Neural Pathway Strength Diagrams (NPSDs) are a novel visualisation technique for viewing the internal operation of 2-layer feedforward neural networks during and following training. This tool facilitates visual inspection of ANNs' weights in a way that is quantitatively precise as well as being visually intuitive. Because it does not summarise weight information, but includes all weight values, it is also particularly useful as a diagnostic technique.

Instead of plotting weights individually, (for example in Hinton diagrams) each neural pathway from input to output of 1HL networks is represented by a single x-y locus within a 2-dimensional neural pathway strength space. This corresponds to the two weight values defining each pathway. The convention of plotting hidden unit weights on the x-axis and output unit weights on the y-axis has been adopted. Because all inputs to a hidden layer neuron use the same single output unit weight value, the loci for the pathways through each hidden unit are arranged in a horizontal row or "sememe". Three breakout views of the same data are also proposed and demonstrated to reveal structure in the neural pathway data. These are: 2-dimensional subspaces organised by output neuron, hidden neuron or input feature, with one sub-plot per output, hidden neuron or input.

In the breakout view by output, the way that each output uses the hidden neuron sememes appears as different vertical displacements of these on each sub-plot. In the breakout view by hidden neuron, there is a single row of symbols (or sememe) in each sub-plot, corresponding to each output. In the

breakout view by input, there are as many loci (symbols) on each sub-plot as there are hidden units, multiplied by the number of outputs. By looking at the shape of the symbol cloud on these sub-plots, it is possible to infer if an input is being used predominantly in an excitatory or inhibitory sense; or if no clear pattern emerges. These breakout views provide a further level of detail below that provided by CNPSA and can allow diagnosis of ANN behaviour.

6.1.5 Multi-output ANNs for urban flood modelling and prediction

The novel use of multi-output ANNs to model urban flooding at multiple sewer nodes or locations simultaneously is extensively demonstrated and evaluated using three urban drainage networks: Crossness (south London), Portsmouth and Dorchester, England. These exploit the similarities in hydrographical responses to rainfall at various locations in an urban drainage network. This means that the same set of hidden layer neurons can be re-used to construct the various ANN responses in a way that is computationally efficient, yet does not degrade performance to an unacceptable extent.

Examples have been shown to work with both design and real rainfall. A challenge connected with modelling spatially variable rainfall in large catchments has been successfully identified. Early solutions to this have also been trialled using partitioned modular models for sub-catchments within the main catchment. It is likely that upstream, midstream and downstream areas of catchments could each be modelled effectively by separate modular models, since they would share even greater commonality in the hydrograph response shapes needed for each.

The application of NPSFS to multi-nodal urban flood modelling has so far not been trialled.

Indications are that ANN tools are generally good and computationally efficient for prediction of flooding in urban drainage systems to a level of accuracy which the water industry would find useful. Model preparation would need further automation in order to facilitate use by water professionals not expert in neural networks and machine learning.

6.1.6 ROC scenarios and neuro-evolution for classifier ensembles

The use of ROC scenarios for the optimisation of ANNs for bathing water quality classifiers has also been extensively researched and demonstrated in this thesis. NFCV has been used to build ensembles of models that perform better collectively than their individual members. The ROC is also used together with neuro-evolution (NSGA-II) in order to facilitate a dual-objective approach to optimising (minimising) both false positive rate (FPR) and false negative rate (FNR) simultaneously. This is compared with the gradient-based SCG method optimising (maximising) the single objective of area under the ROC curve. In practice the SCG method has been found to be more reliable and the models so produced have performed better than the NSGA-II approach used. However, both have produced acceptable results.

To the author's knowledge, the use of ROC, ANN NFCV ensembles and NPSFS are all novel approaches for the application of bathing water quality prediction. The ANN model ensembles have also been compared with decision tree (DT) and simple threshold models. It has been demonstrated that by using an ROC the optimum operating point for ANN models can be chosen taking into account the trade-off between commercial interests and public safety. This provides a distinct advantage over models with single operating points.

The Bacti case study provides an excellent benchmark allowing demonstration of the NFCV / NPSFS approach to selecting input feature reducts based on their degree of relevance, using the EQR metric. The reduced-input models perform either equally well or sometimes better than the original models with the full input set, but enjoy the benefit of being simpler, more parsimonious models. The input feature sets used include 12 features, but there are other potentially readily available data sources, the use of which may lead to further model improvements.

6.2 Future work

The research covered in this thesis opens up a number of key opportunities for potential further exploration in future projects:

The neural pathway strength analysis could be extended to deeper ANNs; for example 2HL feedforward networks. The method would be extensible by successive multiplications of the layer weight matrices. The treatment of multi-output ANNs could also be investigated, to explore how well the EQR / NPSFS feature-selection approach could be extended to these. This would have immediate practical application in the area of urban flood modelling in an extension of the case study work covered in chapter 3.

The neural pathway strength diagrams could potentially be extended to 2HL networks by use of 3D plots, though there would be a combinatorial increase in the number of data-points to be displayed. Deeper networks may be able to be charted using star diagrams.

The use of multi-output ANNs to model and predict flooding in urban drainage networks could be extended to modelling pluvial flooding, by creating surrogates of 2D hydrodynamic surface flooding models. These could potentially be combined with the sewer flood models.

For modelling of the urban drainage network flooding, the use of hybrid modular models could be investigated. The sewer nodes to be modelled could be divided using a number of heuristics, such as by upstream / midstream / downstream location, by shape of hydrograph, by nearest raingauge or by sub-catchment. The relative performances of these would need to be investigated. Additionally, in a move to increase the water engineering uptake of these techniques, possibilities for automating the above meta-modelling approaches could be researched. This would have the potential benefit of simplifying and supporting modelling decisions needed to create live real-time Early Warning Systems (EWS) using these machine-learning tools.

Additionally, the use of EQR / NPSFS could be investigated as a method of optimising the set of lags to use in the moving time windows of lagged-input ANNs, such as those described in chapter 3. The aim of this would be to provide the benefits of using time-lagged inputs, whilst also achieving optimal input feature reduction.

It may also be possible to extend the use of the ANN models to the nowcasting (short-term prediction up to 6-hours) of local rainfall based on rain radar images. This would then potentially allow these models to be cascaded with those already described to provide operationally useful predictions of flooding or bathing water quality. Early results obtained in this regard show promise, but are not yet ready for formal presentation.

For the Bacti modelling, there is considerable potential for future work to explore a number of additional input features, including perhaps, meteorological features, such as octal cloud cover, wind-speed, direction, air temperature, atmospheric pressure, relative humidity, UV levels and predictions of rainfall. Additionally, hydrological / oceanographic parameters could be potentially included, such as wave height, turbidity, sea surface temperature, dissolved oxygen, river flow rates, CSO spill data and salinity of streams (where present). The hope would be that model performance would improve further; reducing both the number of false positives and false negatives.

The Bacti models created to date are for individual beaches, but there is considerable merit to attempting to construct and assess combined models that cover more than one or even many beaches. In the UK alone, there are 608 designated bathing beaches; so strategies for combining models as well as the semi-automation of their production would no doubt be useful. Such combined models may possibly need to employ time-invariant catchment characteristic data as inputs to the models. These could include such factors as catchment total area, percentages under various land uses (including various rural and urban), catchment average steepness, lengths of watercourses, numbers of / distances to CSOs from the beach. Some of this data is already available from "source apportionment" project work already undertaken by / through the Environment Agency.

Finally, the work conducted in this case study has used multi-layer perceptrons (MLPs) with 2-layers of neurons. Use of other types of machine learning models could also be explored. These could include deeper MLP networks, Bayesian Belief Networks (BBNs), Support Vector Machines (SVMs) and/or Relevance Vector Machines. By using NFCV ensembles of these it may

also be possible to look at the variability of their internal parameterisations across ensemble members as a way of also selecting their input features.

References

Abrahart, R.J., Anctil, F., Coulibaly, P., Dawson, C.W., Mount, N.J., See, L.M., Shamseldin, A.Y., Solomatine, D.P., Toth, E., Wilby, R.L., 2012. Two decades of anarchy? Emerging themes and outstanding challenges for neural network river forecasting. Progress in Physical Geography 36, 480–513. doi:10.1177/0309133312444943

Abrahart, R.J., Heppenstall, A.J., See, L.M., 2007. Timing error correction procedure applied to neural network rainfall—runoff modelling. Hydrological Sciences Journal 52, 414–431. doi:10.1623/hysj.52.3.414

Achard, S., Salvador, R., Whitcher, B., Suckling, J., Bullmore, E., 2006. A Resilient, Low-Frequency, Small-World Human Brain Functional Network with Highly Connected Association Cortical Hubs. J. Neurosci. 26, 63–72. doi:10.1523/JNEUROSCI.3874-05.2006

Acuña, G., Cubillos, F., Thibault, J., Latrille, E., 1999. Comparison of methods for training grey-box neural network models. Computers & Chemical Engineering 23, Supplement, S561–S564. doi:10.1016/S0098-1354(99)80138-0

Admiralty, U., 2014. Admiralty EasyTide - Tidal Prediction [WWW Document]. United Kingdom Hydrographic Office. URL http://www.ukho.gov.uk/Easytide/easytide/SelectPort.aspx (accessed 6.25.14).

Aguilera, P.., Frenich, A.G., Torres, J.., Castro, H., Vidal, J.L.M., Canton, M., 2001. Application of the kohonen neural network in coastal water management: methodological development for the assessment and prediction of water quality. Water Research 35, 4053–4062. doi:10.1016/S0043-1354(01)00151-8

Ahmed, S.E., 2008. Markov Chain Monte Carlo: Stochastic Simulation for Bayesian Inference. Technometrics 50, 97–97.

Alexander, G.N., Karoly, A., Susts, A.B., 1969. Equivalent distributions with application to rainfall as an upper bound to flood distributions. Journal of Hydrology 9, 322–344. doi:10.1016/0022-1694(69)90025-0

Amari, S.-I., Murata, N., Muller, K.-R., Finke, M., Yang, H.H., 1997. Asymptotic statistical theory of overtraining and cross-validation. IEEE Transactions on Neural Networks 8, 985–996. doi:10.1109/72.623200

Amin, S.M., Wollenberg, B.F., 2005. Toward a smart grid: power delivery for the 21st century. IEEE Power and Energy Magazine 3, 34–41. doi:10.1109/MPAE.2005.1507024

Amis, G.P., Carpenter, G.A., 2010. Self-supervised ARTMAP. Neural Networks 23, 265–282. doi:10.1016/j.neunet.2009.07.026

Anastasio, M.A., Kupinski, M.A., Nishikawa, R.M., 1998. Optimization and FROC analysis of rule-based detection schemes using a multiobjective approach. IEEE Transactions on Medical Imaging 17, 1089–1093. doi:10.1109/42.746726

Atiquzzaman, M., Liong, S.-Y., Yu, X., 2006. Alternative decision making in water distribution network with NSGA-II. Journal of water resources planning and management 132, 122–126.

Bache, K., Lichman, M., 2013. UCI Repository of machine learning databases [WWW Document]. UCI Machine Learning Repository. URL http://archive.ics.uci.edu/ml/ (accessed 9.24.13).

Bader, J., Zitzler, E., 2010. HypE: An Algorithm for Fast Hypervolume-Based Many-Objective Optimization. Evolutionary Computation 19, 45–76. doi:10.1162/EVCO_a_00009

Barlow, H., 1989. Unsupervised learning. Neural computation 1, 295–311.

Barron, A.R., 1993. Universal approximation bounds for superpositions of a sigmoidal function. Information Theory, IEEE Transactions on 39, 930 –945. doi:10.1109/18.256500

Bartlett, P.L., 1998. The sample complexity of pattern classification with neural networks: the size of the weights is more important than the size of the network. IEEE Transactions on Information Theory 44, 525–536. doi:10.1109/18.661502

Battiti, R., 1992. First-and second-order methods for learning: between steepest descent and Newton's method. Neural computation 4, 141–166.

Beaubouef, T., Petry, F., 2012. Rough and Rough-Fuzzy Sets in Design of Information Systems, in: Meyers, R.A. (Ed.), Computational Complexity. Springer New York, New York, NY, pp. 2702–2715.

Bechikh, S., Said, L.B., Ghédira, K., 2011. Searching for knee regions of the Pareto front using mobile reference points. Soft Comput 15, 1807–1823. doi:10.1007/s00500-011-0694-3

Becker, S., 1991. Unsupervised learning procedures for neural networks. International Journal of Neural Systems 2, 17–33.

Bekele, E.G., Nicklow, J.W., 2007. Multi-objective automatic calibration of SWAT using NSGA-II. Journal of Hydrology 341, 165–176. doi:10.1016/j.jhydrol.2007.05.014
Beraud, B., Lemoine, C., Steyer, J.-P., 2009. Multiobjective genetic algorithms for the optimisation of wastewater treatment processes, in: Computational Intelligence Techniques for Bioprocess Modelling, Supervision and Control. Springer, pp. 163–195.
Bernecker, T., Houle, M., Kriegel, H.P., Kröger, P., Renz, M., Schubert, E., Zimek, A., 2011. Quality of similarity rankings in time series. Advances in Spatial and Temporal Databases 422–440.
Beven, K., 2006. A manifesto for the equifinality thesis. Journal of hydrology 320, 18–36.
Beven, K., Freer, J., 2001. Equifinality, data assimilation, and uncertainty estimation in mechanistic modelling of complex environmental systems using the GLUE methodology. Journal of hydrology 249, 11–29.
Biogest, 2014. Lamella baffle, Type LTW [WWW Document]. www.biogest.com. URL http://www.biogest.com/products/combined-sewer-overflows/lamella-baffle-type-ltw (accessed 4.12.15).
Bishop, C., 2006. Pattern Recognition and Machine Learning. Springer, Berlin.
Bishop, C.H., Toth, Z., 1999. Ensemble Transformation and Adaptive Observations. Journal of the Atmospheric Sciences 56, 1748–1765. doi:10.1175/1520-0469(1999)056<1748:ETAAO>2.0.CO;2
Bishop, C.M., 1995. Neural networks for pattern recognition. OUP, Oxford, UK.
Bowden, G.J., Dandy, G.C., Maier, H.R., 2005. Input determination for neural network models in water resources applications. Part 1—background and methodology. Journal of Hydrology 301, 75–92. doi:10.1016/j.jhydrol.2004.06.021
Bradley, A.P., 1997. The use of the area under the ROC curve in the evaluation of machine learning algorithms. Pattern Recognition 30, 1145–1159. doi:10.1016/S0031-3203(96)00142-2
Branke, J., 1995. Evolutionary Algorithms for Neural Network Design and Training, in: IN PROCEEDINGS OF THE FIRST NORDIC WORKSHOP ON GENETIC ALGORITHMS AND ITS APPLICATIONS. pp. 145–163.
Branke, J., Deb, K., Dierolf, H., Osswald, M., 2004. Finding knees in multi-objective optimization, in: Parallel Problem Solving from Nature-PPSN VIII. pp. 722–731.
Breiman, L., 1996. Bagging predictors. Mach Learn 24, 123–140. doi:10.1007/BF00058655
Brion, G.M., Lingireddy, S., 2003. Artificial neural network modelling: a summary of successful applications relative to microbial water quality. Health-related Water Microbiology 47, 235–240.
Brodmann, K., 1909. Vergleichende Lokalisationslehre der Gro hirnrinde. Springer.
Bruen, M., Yang, J., 2006. Combined Hydraulic and Black-Box Models for Flood Forecasting in Urban Drainage Systems. Journal of Hydrologic Engineering 11, 589–596. doi:10.1061/(ASCE)1084-0699(2006)11:6(589)
Bruner, J.S., 1957. Neural mechanisms in perception. Psychological Review 64, 340.
Buhrman, H., De Wolf, R., 2002. Complexity measures and decision tree complexity: a survey. Theoretical Computer Science 288, 21–43.
Bullmore, E., Sporns, O., 2009. Complex brain networks: graph theoretical analysis of structural and functional systems. Nat Rev Neurosci 10, 186–198. doi:10.1038/nrn2575
Butler, D., Davies, J., 2004. Time of Concentration, in: Urban Drainage. Taylor & Francis, pp. 249–257.
Cameron, D.S., Beven, K.J., Tawn, J., Blazkova, S., Naden, P., 1999. Flood frequency estimation by continuous simulation for a gauged upland catchment (with uncertainty). Journal of Hydrology 219, 169–187. doi:10.1016/S0022-1694(99)00057-8
Campolo, M., 2003. Artificial neural network approach to flood forecasting in the River Arno. Hydrological Sciences, 48(3) 381–398.
Carpenter, G.A., Gjaja, M.N., Gopal, S., Woodcock, C.E., 1997a. ART neural networks for remote sensing: vegetation classification from Landsat TM and terrain data. IEEE Transactions on Geoscience and Remote Sensing 35, 308–325. doi:10.1109/36.563271
Carpenter, G.A., Martens, S., Ogas, O.J., 2005. Self-organizing information fusion and hierarchical knowledge discovery: a new framework using ARTMAP neural networks. Neural Networks 18, 287–295. doi:10.1016/j.neunet.2004.12.003
Carpenter, G.A., Milenova, B.L., Noeske, B.W., 1998. Distributed ARTMAP: a neural network for fast distributed supervised learning. Neural Networks 11, 793–813. doi:10.1016/S0893-6080(98)00019-7
Carpenter, G.A., Rubin, M.A., Streilein, W.W., 1997b. ARTMAP-FD: familiarity discrimination applied to radar target recognition, in: , International Conference on Neural

Networks,1997. Presented at the , International Conference on Neural Networks,1997, pp. 1459–1464 vol.3. doi:10.1109/ICNN.1997.614010
Caruana, R., Lawrence, S., Giles, L., 2001. Overfitting in neural nets: Backpropagation, conjugate gradient, and early stopping. Advances in neural information processing systems 402–408.
Cawley, G.C., Talbot, N.L.C., 2003. Efficient leave-one-out cross-validation of kernel Fisher discriminant classifiers. Pattern Recognition 36, 2585–2592.
Chakraborty, D.P., 1989. Maximum likelihood analysis of free-response receiver operating characteristic (FROC) data. Medical physics 16, 561–568.
Chan, S.N., Thoe, W., Lee, J.H.W., 2013. Real-time forecasting of Hong Kong beach water quality by 3D deterministic model. Water Research 47, 1631–1647. doi:10.1016/j.watres.2012.12.026
Chávez, E., Navarro, G., Baeza-Yates, R., Marroquín, J.L., 2001. Searching in metric spaces. ACM Computing Surveys (CSUR) 33, 273–321.
Chen, H., Yao, X., 2010. Multiobjective Neural Network Ensembles Based on Regularized Negative Correlation Learning. IEEE Transactions on Knowledge and Data Engineering 22, 1738–1751. doi:10.1109/TKDE.2010.26
Chen, S., Hong, X., Harris, C.J., 2011. Grey-box radial basis function modelling. Neurocomputing 74, 1564–1571. doi:10.1016/j.neucom.2011.01.023
Chen, X., Li, Y.S., Liu, Z., Yin, K., Li, Z., Wai, O.W., King, B., 2004. Integration of multi-source data for water quality classification in the Pearl River estuary and its adjacent coastal waters of Hong Kong. Continental Shelf Research 24, 1827–1843. doi:10.1016/j.csr.2004.06.010
Cherkauer, K.J., 1996. Human expert-level performance on a scientific image analysis task by a system using combined artificial neural networks, in: Working Notes of the AAAI Workshop on Integrating Multiple Learned Models. Citeseer, pp. 15–21.
Chiang, Y.-M., Chang, L.-C., Tsai, M.-J., Wang, Y.-F., Chang, F.-J., 2010. Dynamic neural networks for real-time water level predictions of sewerage systems-covering gauged and ungauged sites. Hydrology and Earth System Sciences 14, 1309–1319.
Chu, V.T., 2007. Is reinforcement needed in precast concrete manhole units? [WWW Document]. Civil Engineering Portal. URL http://www.engineeringcivil.com/is-reinforcement-needed-in-precast-concrete-manhole-units.html (accessed 4.12.15).
Ciresan, D., Giusti, A., Schmidhuber, J., 2012. Deep neural networks segment neuronal membranes in electron microscopy images, in: Advances in Neural Information Processing Systems 25. pp. 2852–2860.
Cloke, H.L., Pappenberger, F., 2009. Ensemble flood forecasting: a review. Journal of Hydrology 375, 613–626.
Cochocki, A. and U., 1993. Neural Networks for Optimization and Signal Processing. John Wiley & Sons, Inc., New York NY.
Coello, C.A.C., Pulido, G.T., Lechuga, M.S., 2004. Handling multiple objectives with particle swarm optimisation. IEEE Trans. on Evolutionary Computation. 8(3): 256–279.
Collobert, R., Weston, J., 2008. A unified architecture for natural language processing: Deep neural networks with multitask learning, in: Proceedings of the 25th International Conference on Machine Learning. ACM, pp. 160–167.
Corani, G., Guariso, G., 2005. An application of pruning in the design of neural networks for real time flood forecasting. Neural Comput & Applic 14, 66–77. doi:10.1007/s00521-004-0450-z
Cortez, P., Morais, A., 2008. UCI Forest Fires Data Set [WWW Document]. UCI Machine Learning Repository. URL http://archive.ics.uci.edu/ml/datasets/Forest+Fires (accessed 9.20.13).
Cortez, P., Morais, A., 2007. A Data Mining Approach to Predict Forest Fires using Meteorological Data, in: New Trends in Artificial Intelligence. Presented at the Proceedings of the 13th EPIA 2007 - Portuguese Conference on Artificial Intelligence, University of Minho, Guimaraes, Portugal, pp. 512–523.
Couckuyt, I., Gorissen, D., Rouhani, H., Laermans, E., Dhaene, T., 2009. Evolutionary regression modeling with active learning: An application to rainfall runoff modeling, in: Adaptive and Natural Computing Algorithms. Springer, pp. 548–558.
Cunningham, P., Carney, J., 2000. Diversity versus Quality in Classification Ensembles Based on Feature Selection, in: Mántaras, R.L. de, Plaza, E. (Eds.), Machine Learning: ECML 2000, Lecture Notes in Computer Science. Springer Berlin Heidelberg, pp. 109–116.
Cybenko, G., 1989. Approximation by superpositions of a sigmoidal function. Math. Control Signal Systems 2, 303–314. doi:10.1007/BF02551274

Dash, M., Liu, H., 2003. Consistency-based search in feature selection. Artificial intelligence 151, 155–176.
Das, I., 1999. On characterizing the “knee” of the Pareto curve based on Normal-Boundary Intersection. Structural Optimization 18, 107–115. doi:10.1007/BF01195985
Dastorani, M.T., Moghadamnia, A., Piri, J., Rico-Ramirez, M., 2010. Application of ANN and ANFIS models for reconstructing missing flow data. Environmental monitoring and assessment 166, 421–434.
Dawkins, R., 2006. The selfish gene. Oxford university press.
Dawson, C.W., Wilby, R.L., 2001. Hydrological modelling using artificial neural networks. Progress in Physical Geography 25, 80–108. doi:10.1177/030913330102500104
Deb, K., Anand, A., Joshi, D., 2002a. A Computationally Efficient Evolutionary Algorithm for Real-Parameter Optimization. Evolutionary Computation 10, 371–395. doi:10.1162/106365602760972767
Deb, K., Pratap, A., Agarwal, S., Meyarivan, T., 2002b. A fast and elitist multiobjective genetic algorithm: NSGA-II. Evolutionary Computation, IEEE Transactions on 6, 182–197.
DEFRA, 2013. 2013 mandatory compliance results for bathing waters in the UK (EC rBWD compliance report No. PB14055). Department for Environment, Food and Rural Affairs, London, UK.
De Groot, W.J., 1998. Interpreting the Canadian Forest Fire Weather Index (FWI) System, in: Proc. of the Fourth Central Region Fire Weather Committee Scientific and Technical Seminar.
Delelegn, S.W., Pathirana, A., Gersonius, B., Adeogun, A.G., Vairavamoorthy, K., 2011. Multi-objective optimisation of cost-benefit of urban flood management using a 1 D 2 D coupled model. Water Science and Technology 63, 1054.
Deng, Z., Namwamba, F., Zhang, Z., 2012. Development of Decision Support System for Beach Water Quality Management, in: 10th International Conference on Hydroinformatics. Presented at the 10th International Conference on Hydroinformatics, IWA / IAHR, Hamburg, Germany.
Dhanalakshmi, S., Kannan, S., Mahadevan, K., Baskar, S., 2011. Application of modified NSGA-II algorithm to combined economic and emission dispatch problem. International Journal of Electrical Power & Energy Systems 33, 992–1002.
Diao, R., Shen, Q., 2012. Feature Selection With Harmony Search. IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics 42, 1509–1523. doi:10.1109/TSMCB.2012.2193613
Diao, R., Shen, Q., 2011. Fuzzy-rough classifier ensemble selection, in: 2011 IEEE International Conference on Fuzzy Systems (FUZZ). Presented at the 2011 IEEE International Conference on Fuzzy Systems (FUZZ), pp. 1516–1522. doi:10.1109/FUZZY.2011.6007400
Dibike, Y.B., Solomatine, D., abbott, M.B., 1999. On the encapsulation of numerical-hydraulic models in artificial neural network. Journal of Hydraulic research 37, 147–161.
Dietterich, T.G., 2000. Ensemble methods in machine learning, in: Multiple Classifier Systems. Springer, pp. 1–15.
Di Pierro, F., Khu, S.-T., Savić, D., Berardi, L., 2009. Efficient multi-objective optimal design of water distribution networks on a budget of simulations using hybrid algorithms. Environmental Modelling & Software 24, 202–213.
Dorffner, G., 1996. Neural Networks for Time Series Processing. Neural Network World 6, 447–468.
Dorigo, M., Blum, C., 2005. Ant colony optimization theory: A survey. Theoretical Computer Science 344, 243–278. doi:10.1016/j.tcs.2005.05.020
Duncan, A., Chen, A.S., Keedwell, E., Djordjevic, S., Savic, D.A., 2011. Urban flood prediction in real-time from weather radar and rainfall data using artificial neural networks, in: IAHS Red Book Series No. 351, 58. Presented at the Weather Radar and Hydrology International Symposium, International Association of Hydrological Sciences, Exeter, UK.
Duncan, A.P., Chen, A.S., Keedwell, E.C., Djordjevic, S., Savic, D.A., 2013a. RAPIDS: Early Warning System for Urban Flooding and Water Quality Hazards (Extended Abstract), in: AISB 2013. Presented at the Artificial Intelligence and Simulation of Behaviour Conference; Machine Learning in Water Systems Symposium, AISB, University of Exeter, pp. 25–29.
Duncan, A.P., Keedwell, E.C., Djordjevic, S., Savic, D.A., 2013b. Machine Learning-Based Early Warning System for Urban Flood Management, in: International Conference on

Flood Resilience: Experiences in Asia and Europe. Presented at the International Conference on Flood Resilience 2013, University of Exeter, Exeter, UK, pp. 237–238.
Duncan, A.P., Keedwell, E.C., Djordjevic, S., Savic, D.A., 2013c. Early Warning System for Bathing Water Quality (Poster), in: Bathing Waters 2013. Presented at the Bathing Waters 2013, Defra, England, Southport, UK.
Duncan, A.P., Tyrrell, D., Smart, N., Keedwell, E.C., Djordjevic, S., Savic, D.A., 2013d. Comparison of machine learning classifier models for bathing water quality exceedances in UK, in: IAHR35. Presented at the IAHR35, IAHR, Chengdu, China.
Einfalt, T., Arnbjerg-Nielsen, K., Golz, C., Jensen, N.-E., Quirmbach, M., Vaes, G., Vieux, B., 2004. Towards a roadmap for use of radar rainfall data in urban drainage. Journal of Hydrology 299 186–202.
Elman, J.L., 1991. Distributed representations, simple recurrent networks, and grammatical structure. Machine learning 7, 195–225.
Elman, J.L., 1990. Finding structure in time. Cognitive science 14, 179–211.
Environment Agency, 2015a. Bathing Water Profile for East Looe [WWW Document]. Environment Agency. URL http://environment.data.gov.uk/bwq/profiles/profile.html?_search=east%20looe&site=ukk3101-27000 (accessed 4.12.15).
Environment Agency, 2015b. Bathing Water Profile for Seaton (Cornwall) [WWW Document]. Environment Agency. URL http://environment.data.gov.uk/bwq/profiles/profile.html?_search=seaton&site=ukk3101-26800 (accessed 4.12.15).
Environment Agency, 2015c. Bathing Water Profile for Readymoney [WWW Document]. Environment Agency. URL http://environment.data.gov.uk/bwq/profiles/profile.html?_search=readym&site=ukk3106-27100 (accessed 4.12.15).
Environment Agency, 2015d. Bathing Water Profile for Par [WWW Document]. Environment Agency. URL http://environment.data.gov.uk/bwq/profiles/profile.html?_search=par&site=ukk3106-27300 (accessed 4.12.15).
Environment Agency, 2015e. Bathing Water Profile for Porthluney [WWW Document]. Environment Agency. URL http://environment.data.gov.uk/bwq/profiles/profile.html?_search=porthlu&site=ukk3106-28400 (accessed 4.12.15).
European Commission, 2006a. Revised Bathing Water Directive (2006/7/EC).
European Commission, 2006b. European SmartGrids technology platform: vision and strategy for europe's electricity networks of the future. Directorate for Research EUR 22040.
European Commission, 1976. 76/160/EEC of 8 December 1975 concerning the quality of bathing water. OJ L 31.
Everson, R.M., Fieldsend, J.E., 2006. Multiobjective optimization of safety related systems: an application to short-term conflict alert. IEEE Transactions on Evolutionary Computation 10, 187 – 198. doi:10.1109/TEVC.2005.856067
Faulkner, D., 1999. Flood estimation handbook, volume 2: Rainfall frequency estimation. Institute of Hydrology Wallingford.
Fawcett, T., 2006. An introduction to ROC analysis. Pattern Recognition Letters, ROC Analysis in Pattern Recognition 27, 861–874. doi:10.1016/j.patrec.2005.10.010
Fay, M.P., Proschan, M.A., 2010. Wilcoxon-Mann-Whitney or t-test? On assumptions for hypothesis tests and multiple interpretations of decision rules. Stat Surv 4, 1–39. doi:10.1214/09-SS051
Fernando, A., 2005. Combined Sewer Overflow forecasting with Feed-forward Back-propagation Artificial Neural Network. International Journal of Applied Science, Engineering and Technology 1;4 211–217.
Fernando, T., Maier, H.R., Dandy, G.C., May, R., 2005. Efficient selection of inputs for artificial neural network models, in: Proc. of MODSIM 2005 International Congress on Modelling and Simulation: Modelling and Simulation Society of Australia and New Zealand.
Ferreira, J.C., Fonseca, C.M., Gaspar-Cunha, A., 2007. Methodology to select solutions from the pareto-optimal set: a comparative study, in: Proceedings of the 9th Annual Conference on Genetic and Evolutionary Computation, GECCO '07. ACM, New York, NY, USA, pp. 789–796. doi:10.1145/1276958.1277117
Fieldsend, J.E., Everson, R.M., 2002. ROC optimisation of safety related systems. database 20701, 0.

Figueiredo, M.A., Jain, A.K., 2002. Unsupervised learning of finite mixture models. Pattern Analysis and Machine Intelligence, IEEE Transactions on 24, 381–396.
Fodor, I.K., 2002. A Survey of Dimension Reduction Techniques. U.S. Department of Energy Office of Scientific and Technical Information.
Franklin, J.A., 1989. Historical perspective and state of the art in connectionist learning control, in: , Proceedings of the 28th IEEE Conference on Decision and Control, 1989. Presented at the , Proceedings of the 28th IEEE Conference on Decision and Control, 1989, pp. 1730–1736 vol.2. doi:10.1109/CDC.1989.70447
Freund, Y., 1995. Boosting a Weak Learning Algorithm by Majority. Information and Computation 121, 256–285. doi:10.1006/inco.1995.1136
Freund, Y., Schapire, R.E., 1996. Experiments with a new boosting algorithm, in: ICML. pp. 148–156.
FRMRC2, 2011. Flood Risk Management Research Consortium Website.
Fu, G., Butler, D., Khu, S.-T., 2008. Multiple objective optimal control of integrated urban wastewater systems. Environmental Modelling & Software 23, 225–234.
Fu, G., Khu, S.-T., Butler, D., 2009. Use of surrogate modelling for multiobjective optimisation of urban wastewater systems.
Gallant, S.I., 1988. Connectionist expert systems. Commun. ACM 31, 152–169. doi:10.1145/42372.42377
Garbrecht, J., 2006. Comparison of Three Alternative ANN Designs for Monthly Rainfall-Runoff Simulation. Journal of Hydrologic Engineering 11, 502–505. doi:10.1061/(ASCE)1084-0699(2006)11:5(502)
Garson, G.D., 1991. Interpreting Neural-network Connection Weights. AI Expert 6, 46–51.
Goldberg, D.E., Holland, J.H., 1988. Genetic Algorithms and. Machine Learning 3, 95–99. doi:10.1023/A:1022602019183
Goswami, M., O'Connor, K.M., 2007. Real-time flow forecasting in the absence of quantitative precipitation forecasts: a multi-model approach. Journal of hydrology 334, 125–140.
Gradštejn, I.S., 2000. Table of integrals, series, and products. Academic Press, San Diego [u.a.
Granger, C.W., 1969. Investigating causal relations by econometric models and cross-spectral methods. Econometrica: Journal of the Econometric Society 424–438.
Grayman, W.M., Deininger, R.A., Males, R.M., 2001. Design of Early Warning and Predictive Source-Water Monitoring Systems. American Water Works Association.
Grossberg, S., 1976. Adaptive pattern classification and universal recoding: I. Parallel development and coding of neural feature detectors. Biol. Cybernetics 23, 121–134. doi:10.1007/BF00344744
Grum, M., Longin, E., Linde, J.J., 2004. A Flexible and Extensible Open Source Tool for Urban Drainage. Modelling: www. WaterAspects. org.
Guo, Z., Uhrig, R.E., 1992. Using genetic algorithms to select inputs for neural networks, in: Combinations of Genetic Algorithms and Neural Networks, 1992., COGANN-92. International Workshop on. IEEE, pp. 223–234.
Guyon, I., Elisseeff, A., 2003. An introduction to variable and feature selection. The Journal of Machine Learning Research 3, 1157–1182.
Habib, E., Krajewski, W., Kruger, A., 2001. Sampling Errors of Tipping-Bucket Rain Gauge Measurements. Journal of Hydrologic Engineering 6, 159–166. doi:10.1061/(ASCE)1084-0699(2001)6:2(159)
Hall, M.A., 1999. Correlation-based feature selection for machine learning. The University of Waikato.
Han, D., Kwong, T., Li, S., 2007. Uncertainties in real-time flood forecasting with neural networks. Hydrological Processes 21, 223–228. doi:10.1002/hyp.6184
Han, J., 2003. Application of artificial neural networks for flood warning systems.
Hansen, L.K., Salamon, P., 1990. Neural network ensembles. IEEE Transactions on Pattern Analysis and Machine Intelligence 12, 993–1001. doi:10.1109/34.58871
Hapsari, R.I., Oishi, S., Sunada, K., Sano, T., Sisinggih, D., 2011. Ensemble short-term rainfall—runoff prediction and its application in urban flood risk mapping, in: IAHS-AISH Publication. Presented at the International Conference on Flood Management, International Association of Hydrological Sciences, pp. 308–319.
Harman, H.H., 1960. Modern factor analysis.
Harmon, L.D., 1961. Studies with artificial neurons, I: Properties and functions of an artificial neuron. Biological Cybernetics 1, 89–101.
Harmon, L.D., 1959. Artificial neuron. Science 129, 962–963.
Hassibi, B., Stork, D.G., 1993. Second order derivatives for network pruning: Optimal brain surgeon. Advances in neural information processing systems 164–164.

Hastings, W.K., 1970. Monte Carlo sampling methods using Markov chains and their applications. Biometrika 57, 97–109. doi:10.1093/biomet/57.1.97
Hava T. Siegelmann, E.D.S., 1991. Turing computability with neural nets. Appl. Math. Lett. Vol. ?? 1–15.
Hawkins, D.M., 2004. The problem of overfitting. Journal of chemical information and computer sciences 44, 1–12.
Hebb, D.O., 1955. Drives and the CNS (conceptual nervous system). Psychological review 62, 243.
Hebb, D.O., 1949. The first stage of perception: Growth of the assembly. The Organization of Behavior 60–78.
Hecht-Nielsen, R., 1989. Theory of the backpropagation neural network. pp. 593–605.
Hedar, A.-R., Wang, J., Fukushima, M., 2008. Tabu search for attribute reduction in rough set theory. Soft Computing 12, 909–918.
Heilig, G.K., 2012. World Urbanization Prospects The 2011 Revision. Presentation at the Center for Strategic and International Studies (CSIS) June, Washington, DC.
He, J., Valeo, C., Chu, A., Neumann, N.F., 2011. Prediction of event-based stormwater runoff quantity and quality by ANNs developed using PMI-based input selection. Journal of Hydrology.
Herrera, M., Torgo, L., Izquierdo, J., Pérez-García, R., 2010. Predictive models for forecasting hourly urban water demand. Journal of Hydrology 387, 141–150. doi:10.1016/j.jhydrol.2010.04.005
Hiltz, F.F., 1963. Artificial neuron. Kybernetik 1, 231–236.
Hinton, G., Deng, L., Yu, D., Dahl, G.E., Mohamed, A., Jaitly, N., Senior, A., Vanhoucke, V., Nguyen, P., Sainath, T.N., others, 2012. Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups. Signal Processing Magazine, IEEE 29, 82–97.
Hinton, G.E., 1984. Distributed representations.
Hinton, G.E., Plaut, D.C., Shallice, T., 1993. Simulating brain-damage.
Hofmann, T., 2001. Unsupervised learning by probabilistic latent semantic analysis. Machine learning 42, 177–196.
Holland, J.H., 1975. Adaptation in natural and artificial systems: An introductory analysis with applications to biology, control, and artificial intelligence. U Michigan Press.
Hornik, K., Stinchcombe, M., White, H., 1989. Multilayer feedforward networks are universal approximators. Neural Networks 2, 359–366. doi:10.1016/0893-6080(89)90020-8
Houle, M., Kriegel, H.-P., Kröger, P., Schubert, E., Zimek, A., 2010. Can Shared-Neighbor Distances Defeat the Curse of Dimensionality? Scientific and Statistical Database Management, in: Lecture Notes in Computer Science. Springer Berlin / Heidelberg, pp. 482–500.
Hsieh, W.W., 2001. Nonlinear principal component analysis by neural networks. Tellus A 53, 599–615. doi:10.1034/j.1600-0870.2001.00251.x
Hsu, C.-W., Chang, C.-C., Lin, C.-J., undefined, others, 2003. A practical guide to support vector classification.
Huang, N.E., Shen, Z., Long, S.R., Wu, M.C., Shih, H.H., Zheng, Q., Yen, N.-C., Tung, C.C., Liu, H.H., 1998. The empirical mode decomposition and the Hilbert spectrum for nonlinear and non-stationary time series analysis. Proceedings of the Royal Society of London. Series A: Mathematical, Physical and Engineering Sciences 454, 903–995.
Hubel, D.H., Wiesel, T.N., 1963. Shape and arrangement of columns in cat's striate cortex. The Journal of physiology 165, 559–568.
Hubel, D.H., Wiesel, T.N., 1961. Integrative action in the cat's lateral geniculate body. The Journal of Physiology 155, 385–398.
Hubel, D.H., Wiesel, T.N., 1959. Receptive fields of single neurones in the cat's striate cortex. The Journal of physiology 148, 574.
Hung, T.-C., Chan, K.-Y., 2013. Uncertainty quantifications of Pareto optima in multiobjective problems. J. Intell. Manuf. 24, 385–395. doi:10.1007/s10845-011-0602-9
IBM, C., 2011. IBM SPSS Decision Trees 20 User Manual.
Innovyze, 2012. InfoWorks CS. Innovyze.
Iqbal, J., Guria, C., 2009. Optimization of an operating domestic wastewater treatment plant using elitist non-dominated sorting genetic algorithm. Chemical Engineering Research and Design 87, 1481–1496.
Ishibuchi, H., Tsukamoto, N., Nojima, Y., 2008. Evolutionary many-objective optimization: A short review, in: IEEE Congress on Evolutionary Computation, 2008. CEC 2008. (IEEE World Congress on Computational Intelligence). Presented at the IEEE Congress on

Evolutionary Computation, 2008. CEC 2008. (IEEE World Congress on Computational Intelligence), pp. 2419–2426. doi:10.1109/CEC.2008.4631121
Ivakhnenko, A.G., 1971. Polynomial Theory of Complex Systems. IEEE Transactions on Systems, Man and Cybernetics SMC-1, 364–378. doi:10.1109/TSMC.1971.4308320
Jacobs, R.A., 1988. Increased rates of convergence through learning rate adaptation. Neural Networks 1, 295–307. doi:10.1016/0893-6080(88)90003-2
Jain, A., Sudheer, K.P., Srinivasulu, S., 2004. Identification of physical processes inherent in artificial neural network rainfall runoff models. Hydrological Processes 18, 571–581. doi:10.1002/hyp.5502
JamesMadisonUniversity, 2012. General Evolutionary Algorithm block diagram [WWW Document]. James Madison University. URL http://www.jmu.edu/geology/ComplexEvolutionarySystems/ElaboratingEvolution.html
Jensen, R., Cornelis, C., 2008. A new approach to fuzzy-rough nearest neighbour classification, in: Rough Sets and Current Trends in Computing. pp. 310–319.
Jensen, R., Shen, Q., 2009. New Approaches to Fuzzy-Rough Feature Selection. IEEE Transactions on Fuzzy Systems 17, 824–838. doi:10.1109/TFUZZ.2008.924209
Jin, Y., Gruna, R., Sendhoff, B., 2009. Pareto analysis of evolutionary and learning systems. Frontiers of Computer Science in China 3, 4–17.
Jin, Y., Okabe, T., Sendhoff, B., 2004. Neural network regularization and ensembling using multi-objective evolutionary algorithms, in: Congress on Evolutionary Computation, 2004. CEC2004. Presented at the Congress on Evolutionary Computation, 2004. CEC2004, pp. 1–8 Vol.1. doi:10.1109/CEC.2004.1330830
Jolliffe, I., 2005. Principal component analysis. Wiley Online Library.
Jordan, M.I., Rumelhart, D.E., 1992. Forward Models: Supervised Learning with a Distal Teacher. Cognitive Science 16, 307–354. doi:10.1207/s15516709cog1603_1
Jozefowiez, N., Semet, F., Talbi, E.-G., 2006. Enhancements of NSGA II and its application to the vehicle routing problem with route balancing, in: Artificial Evolution. Springer, pp. 131–142.
Kannan, S., Baskar, S., McCalley, J.D., Murugan, P., 2009. Application of NSGA-II algorithm to generation expansion planning. Power Systems, IEEE Transactions on 24, 454–461.
Kay, D., Bartram, J., Prüss, A., Ashbolt, N., Wyer, M.D., Fleisher, J.M., Fewtrell, L., Rogers, A., Rees, G., 2004. Derivation of numerical values for the World Health Organization guidelines for recreational waters. Water Research 38, 1296–1304. doi:10.1016/j.watres.2003.11.032
Kearns, M., 1988. Thoughts on hypothesis boosting. Unpublished manuscript 45, 105.
Kellagher, R., 2012a. The Use of Artificial Neural Networks (ANNs) in Modelling Sewerage Systems for Management in Real Time: Volume 1 - UKWIR Main Report (12/SW/01/2).
Kellagher, R., 2012b. Preliminary rainfall runoff management for developments-R&D Technical Report (Technical No. W5-074/A/TR/1). HR Wallingford, Wallingford, UK.
Kennedy, J., Eberhart, R., 1995. Particle swarm optimization, in: Neural Networks, 1995. Proceedings., IEEE International Conference on. pp. 1942–1948.
Khashei, M., Bijari, M., 2011. A novel hybridization of artificial neural networks and ARIMA models for time series forecasting. Applied Soft Computing 11, 2664–2675. doi:10.1016/j.asoc.2010.10.015
Khu, S.-T., Keedwell, E., 2005. Introducing more choices (flexibility) in the upgrading of water distribution networks: the New York city tunnel network example. Engineering optimization 37, 291–305.
Khu, S.T., Madsen, H., 2005. Multiobjective calibration with Pareto preference ordering: An application to rainfall-runoff model calibration. Water Resources Research 41.
Khu, S.T., Madsen, H., 2003. A new approach to multi-criteria calibration of rainfall-runoff model, in: Proceedings of the International Conference on Water and Environment: Watershed Hydrology, Allied Publishers, New Delhi, India. pp. 307–316.
Kim On, C., Teo, J., 2010. Evolution and analysis of self-synthesized minimalist neural controllers for collective robotics using Pareto multi-objective optimization, in: 2010 IEEE Congress on Evolutionary Computation (CEC). Presented at the 2010 IEEE Congress on Evolutionary Computation (CEC), pp. 1–7. doi:10.1109/CEC.2010.5586537
Kim, T., Heo, J., Bae, D., Kim, J., 2008. Single-reservoir operating rules for a year using multiobjective genetic algorithm. Journal of Hydroinformatics 10, 163–179.
Kim, T., Heo, J.-H., 2006. Application of multi-objective genetic algorithms to multireservoir system optimization in the Han River basin. KSCE Journal of Civil Engineering 10, 371–380.

Kira, K., Rendell, L.A., 1992a. A Practical Approach to Feature Selection, in: Proceedings of the Ninth International Workshop on Machine Learning, ML '92. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, pp. 249–256.
Kira, K., Rendell, L.A., 1992b. The feature selection problem: Traditional methods and a new algorithm, in: AAAI. pp. 129–134.
Kjeldsen, T.R., 2007. The revitalised FSR/FEH rainfall-runoff method.
Kohavi, R., 1995. A study of cross-validation and bootstrap for accuracy estimation and model selection, in: IJCAI. pp. 1137–1145.
Kohavi, R., John, G.H., 1996. Wrappers for Feature Subset Selection.
Kononenko, I., 1994. Estimating attributes: Analysis and extensions of RELIEF, in: Bergadano, F., Raedt, L.D. (Eds.), Machine Learning: ECML-94, Lecture Notes in Computer Science. Springer Berlin Heidelberg, pp. 171–182.
Kononenko, I., 1990. Comparison of Inductive and Naive Bayesian Learning Approaches to Automatic Knowledge Acquisition, in: Current Trends in Knowledge Acquisition. IOS Press, pp. 190–196.
Kononenko, I., Šimec, E., Robnik-Šikonja, M., 1997. Overcoming the Myopia of Inductive Learning Algorithms with RELIEFF. Applied Intelligence 7, 39–55. doi:10.1023/A:1008280620621
Krasnogor, N., Smith, J., 2005. A tutorial for competent memetic algorithms: model, taxonomy, and design issues. IEEE Transactions on Evolutionary Computation 9, 474–488. doi:10.1109/TEVC.2005.850260
Kuncheva, L.I., Whitaker, C.J., 2003. Measures of Diversity in Classifier Ensembles and Their Relationship with the Ensemble Accuracy. Machine Learning 51, 181–207. doi:10.1023/A:1022859003006
Langley, P., 1996. Elements of machine learning. Morgan Kaufmann.
Lapedes, A. and F.R., 1987. Non-Linear Signal Processing Using Neural Networks. IEEE Neural Networks 1–50.
Larkworthy, T., 2013. The Main Trick In Machine Learning. Edinburgh Hacklab.
Larrañaga, P., Kuijpers, C.M.H., Murga, R.H., Inza, I., Dizdarevic, S., 1999. Genetic algorithms for the travelling salesman problem: A review of representations and operators. Artificial Intelligence Review 13, 129–170.
Lee, K.S., Geem, Z.W., 2005. A new meta-heuristic algorithm for continuous engineering optimization: harmony search theory and practice. Computer methods in applied mechanics and engineering 194, 3902–3933.
Lewis, F.W., Jagannathan, S., Yesildirak, A., 1998. Neural network control of robot manipulators and non-linear systems. CRC Press.
Liang, N.-Y., Huang, G.-B., Saratchandran, P., Sundararajan, N., 2006. A fast and accurate online sequential learning algorithm for feedforward networks. Neural Networks, IEEE Transactions on 17, 1411–1423.
Liang, Y., Liang, X., 2006. Improving signal prediction performance of neural networks through multiresolution learning approach. IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics 36, 341–352. doi:10.1109/TSMCB.2005.857092
Lin, B., Syed, M., Falconer, R.A., 2008. Predicting faecal indicator levels in estuarine receiving waters – An integrated hydrodynamic and ANN modelling approach. Environmental Modelling & Software 23, 729–740. doi:10.1016/j.envsoft.2007.09.009
Liu, H., Yu, L., 2005. Toward integrating feature selection algorithms for classification and clustering. IEEE Transactions on Knowledge and Data Engineering 17, 491–502. doi:10.1109/TKDE.2005.66
Liu, W., Gopal, S., Woodcock, C.E., 2001. Spatial Data Mining with ARTMAP Neural Network, in: Kamath, C., Kegelmeyer, P., Kumar, V., Namburu, R. (Eds.), Data Mining for Scientific and Engineering Applications. Springer, pp. 201–221.
Liu, Y., Yao, X., 1999a. Ensemble learning via negative correlation. Neural Netw 12, 1399–1404.
Liu, Y., Yao, X., 1999b. Simultaneous training of negatively correlated neural networks in an ensemble. IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics 29, 716–725. doi:10.1109/3477.809027
Liu, Y., Yao, X., Higuchi, T., 2000. Evolutionary ensembles with negative correlation learning. IEEE Transactions on Evolutionary Computation 4, 380–387. doi:10.1109/4235.887237
Luk, K.C., Ball, J.E., Sharma, A., 2000. A study of optimal model lag and spatial inputs to artificial neural network for rainfall forecasting. Journal of Hydrology 227, 56–65. doi:10.1016/S0022-1694(99)00165-1

MacKay, D.J., 1995. Probable networks and plausible predictions-a review of practical Bayesian methods for supervised neural networks. Network: Computation in Neural Systems 6, 469–505.
MacKay, D.J.C., Neal, R.M., 1994. Automatic relevance determination for neural networks, in: Technical Report in Preparation. Cambridge University.
Maclin, R., Opitz, D., 2011. Popular Ensemble Methods: An Empirical Study (arXiv e-print No. 1106.0257).
Magnusson, L., 2012. Initial ensemble perturbations - basic concepts.
Maier, H.R., Dandy, G.C., 2000. Neural networks for the prediction and forecasting of water resources variables: a review of modelling issues and applications. Environmental Modelling & Software 15, 101–124. doi:10.1016/S1364-8152(99)00007-9
Maimone, M., Crockett, C., Cesanek, W., 2007. PhillyRiverCast: A Real-Time Bacteria Forecasting Model and Web Application for the Schuylkill River. Journal of Water Resources Planning and Management 133, 542–549. doi:10.1061/(ASCE)0733-9496(2007)133:6(542)
Mansilha, C.R., Coelho, C.A., Heitor, A.M., Amado, J., Martins, J.P., Gameiro, P., 2009. Bathing waters: New directive, new standards, new quality approach. Marine Pollution Bulletin 58, 1562–1565. doi:10.1016/j.marpolbul.2009.03.018
Marquardt, D.W., 1963. An Algorithm for Least-Squares Estimation of Nonlinear Parameters. Journal of the Society for Industrial and Applied Mathematics 11, 431–441. doi:10.1137/0111030
Mathworks, T., 2012. MATLAB® & Simulink® Release Notes for R2012a.
May, R.J., Dandy, G.C., Maier, H.R., Nixon, J.B., 2008. Application of partial mutual information variable selection to ANN forecasting of water quality in water distribution systems. Environmental Modelling & Software 23, 1289–1299.
McCaffrey, J.D., 2014. Evolutionary Optimization using C#. James D. McCaffrey.
McCulloch, W.S. and P., 1943. A logical calculus of the ideas immanent in nervous activity. Bulletin of Mathematical Biophysics Vol 5 115–133.
McKay, M.D., Beckman, R.J., Conover, W.J., 1979. A Comparison of Three Methods for Selecting Values of Input Variables in the Analysis of Output from a Computer Code. Technometrics 21, 239–245. doi:10.2307/1268522
McPhail, C.D., Stidson, R.T., 2009. Bathing water signage and predictive water quality models in Scotland. Aquatic Ecosystem Health & Management 12, 183–186.
MechanicalForex.com, 2014. Artificial Neural Network [WWW Document]. Mechanical Forex. URL http://mechanicalforex.com/wp-content/uploads/2011/06/NN.png (accessed 4.12.15).
Mecklenburg, S., Joss, J., Schmid, W., 2000. Improving the nowcasting of precipitation in an Alpine region with an enhanced radar echo tracking algorithm. Journal of Hydrology 239, 46–68. doi:10.1016/S0022-1694(00)00352-8
Michalewicz, Z., 1996. Genetic algorithms+ data structures= evolution programs. springer.
Millie, D.F., Weckman, G.R., Young II, W.A., Ivey, J.E., Carrick, H.J., Fahnenstiel, G.L., 2012. Modeling microalgal abundance with artificial neural networks: Demonstration of a heuristic “Grey-Box” to deconvolve and quantify environmental influences. Environmental Modelling & Software 38, 27–39. doi:10.1016/j.envsoft.2012.04.009
Miyamoto, H., Kawato, M., Setoyama, T., Suzuki, R., 1988. Feedback-error-learning neural network for trajectory control of a robotic manipulator. Neural Networks 1, 251–265.
Mola, F., 1998. Classification and Regression Trees Software and New Developments, in: Rizzi, A., Vichi, M., Bock, H.-H. (Eds.), Advances in Data Science and Classification, Studies in Classification, Data Analysis, and Knowledge Organization. Springer Berlin Heidelberg, pp. 311–318.
Møller, M.F., 1993. A scaled conjugate gradient algorithm for fast supervised learning. Neural Networks 6, 525–533. doi:10.1016/S0893-6080(05)80056-5
Moody, J.E., Hanson, S.J., Krogh, A., Hertz, J.A., 1995. A simple weight decay can improve generalization. Advances in neural information processing systems 4, 950–957.
Moody, J., Hanson, S.J., Lippmann, R.P., 1992. The Effective Number of Parameters: An Analysis of Generalization and Regularization in Nonlinear Learning Systems. Advances in neural information processing systems 4, 847–854.
Moriasi, D.N., Arnold, J.G., Van Liew, M.W., Bingner, R.L., Harmel, R.D., Veith, T.L., 2007. Model evaluation guidelines for systematic quantification of accuracy in watershed simulations. Transactions of the Asabe 50, 885–900.
Mounce, S., Boxall, J., Machell, J., 2010. Development and Verification of an Online Artificial Intelligence System for Detection of Bursts and Other Abnormal Flows. Journal of Water

Resources Planning and Management 136, 309–318. doi:10.1061/(ASCE)WR.1943-5452.0000030
Mounce, S.R., Khan, A., Wood, A.S., Day, A.J., Widdop, P.D., Machell, J., 2003. Sensor-fusion of hydraulic data for burst detection and location in a treated water distribution system. Information Fusion 4, 217–229. doi:10.1016/S1566-2535(03)00034-4
Mounce, S.R., Mounce, R.B., Boxall, J.B., 2011. Novelty detection for time series data analysis in water distribution systems using support vector machines. Journal of hydroinformatics 13, 672–686.
Mukerji, A., Chatterjee, C., Raghuwanshi, N., 2009. Flood Forecasting Using ANN, Neuro-Fuzzy, and Neuro-GA Models. Journal of Hydrologic Engineering 14, 647–652. doi:10.1061/(ASCE)HE.1943-5584.0000040
Muni, D.P., Pal, N.R., Das, J., 2006. Genetic programming for simultaneous feature selection and classifier design. Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on 36, 106–117.
Nandasana, A.D., Ray, A.K., Gupta, S.K., 2003. Applications of the non-dominated sorting genetic algorithm (NSGA) in chemical reaction engineering. International Journal of Chemical Reactor Engineering 1, No–pp.
Napolitano, G., 2011. An exploration of neural networks for real-time flood forecasting (phd). University of Leeds.
Nash, J.E., Sutcliffe, J.V., 1970. River flow forecasting through conceptual models part I — A discussion of principles. Journal of Hydrology 10, 282–290. doi:10.1016/0022-1694(70)90255-6
Nazemi, A., Chan, A.H., Yao, X., 2008. Selecting representative parameters of rainfall-runoff models using multi-objective calibration results and a fuzzy clustering algorithm, in: BHS 10th National Hydrology Symposium, Exeter. pp. 13–20.
Nazemi, A., Yao, X., Chan, A.H., 2006. Extracting a set of robust Pareto-optimal parameters for hydrologic models using NSGA-II and SCEM, in: Evolutionary Computation, 2006. CEC 2006. IEEE Congress on. IEEE, pp. 1901–1908.
Nguyen, D., Widrow, B., 1990. Improving the learning speed of 2-layer neural networks by choosing initial values of the adaptive weights, in: , 1990 IJCNN International Joint Conference on Neural Networks, 1990. Presented at the , 1990 IJCNN International Joint Conference on Neural Networks, 1990, pp. 21–26 vol.3. doi:10.1109/IJCNN.1990.137819
Norton, R.M., 1984. The double exponential distribution: Using calculus to find a maximum likelihood estimator. The american statistician 38, 135–136.
Oja, E., 1992. Principal components, minor components, and linear neural networks. Neural Networks 5, 927–935. doi:10.1016/S0893-6080(05)80089-9
Oja, E., 1989. Neural networks, principal components, and subspaces. International Journal of Neural Systems 01, 61–68. doi:10.1142/S0129065789000475
Olden, J.D., Jackson, D.A., 2002. Illuminating the “black box”: a randomization approach for understanding variable contributions in artificial neural networks. Ecological Modelling 154, 135–150. doi:10.1016/S0304-3800(02)00064-9
Oliver, I.M., Smith, Dj., Holland, J.R., 1987. Study of permutation crossover operators on the traveling salesman problem, in: Genetic Algorithms and Their Applications: Proceedings of the Second International Conference on Genetic Algorithms: July 28-31, 1987 at the Massachusetts Institute of Technology, Cambridge, MA.
Ortiz-Boyer, D., 2005. Evolutionary Algorithms [WWW Document]. CIXL2: A Crossover Operator for Evolutionary Algorithms Based on Population Features. URL http://www.cs.cmu.edu/afs/cs/project/jair/pub/volume24/ortizboyer05a-html/node7.html (accessed 8.27.14).
Pal, M., Mather, P.M., 2003. An assessment of the effectiveness of decision tree methods for land cover classification. Remote sensing of environment 86, 554–565.
Parzen, E., 1962. On estimation of a probability density function and mode. The annals of mathematical statistics 1065–1076.
Penedo, M., Souto, M., Tahoces, P.G., Carreira, J.M., Villalón, J., Porto, G., Seoane, C., Vidal, J.J., Berbaum, K.S., Chakraborty, D.P., Fajardo, L.L., 2005. Free-Response Receiver Operating Characteristic Evaluation of Lossy JPEG2000 and Object-based Set Partitioning in Hierarchical Trees Compression of Digitized Mammograms [1]. Radiology 237, 450–457. doi:10.1148/radiol.2372040996
Penny, W.D., Roberts, S.J., 1999. Bayesian neural networks for classification: how useful is the evidence framework? Neural Networks 12, 877–892.

Pham, M.-T., Cham, T.-J., 2007. Online learning asymmetric boosted classifiers for object detection, in: Computer Vision and Pattern Recognition, 2007. CVPR'07. IEEE Conference on. IEEE, pp. 1–8.
Pilgrim, D., 2001. AUSTRALIAN RAINFALL AND RUNOFF: VOLUME ONE: A GUIDE TO FLOOD ESTIMATION.
Plumb, A.P., Rowe, R.C., York, P., Brown, M., 2005. Optimisation of the predictive ability of artificial neural network (ANN) models: A comparison of three ANN programs and four classes of training algorithm. European journal of pharmaceutical sciences 25, 395–405.
Preis, A., Ostfeld, A., 2006. Multiobjective sensor design for water distribution systems security, in: 8th Annual Symp. on Water Distribution Systems Analysis. Environmental and Water Resources Institute of ASCE (EWRI of ASCE) New York, pp. 1–17.
Press, W.H., 2007. Numerical Recipes 3rd Edition: The Art of Scientific Computing. Cambridge University Press.
Pyayt, A.L., Mokhov, I.I., Kozionov, A., Kusherbaeva, V., Melnikova, N.B., Krzhizhanovskaya, V.V., Meijer, R.J., 2011a. Artificial intelligence and finite element modelling for monitoring flood defence structures, in: Environmental Energy and Structural Monitoring Systems (EESMS), 2011 IEEE Workshop on. pp. 1–7.
Pyayt, A.L., Mokhov, I.I., Lang, B., Krzhizhanovskaya, V.V., Meijer, R.J., 2011b. Machine Learning Methods for Environmental Monitoring and Flood Protection. World Academy of Science, Engineering and Technology 118–124.
Rachmawati, L., Srinivasan, D., 2009. Multiobjective Evolutionary Algorithm With Controllable Focus on the Knees of the Pareto Front. IEEE Transactions on Evolutionary Computation 13, 810–824. doi:10.1109/TEVC.2009.2017515
Rafiq, M.Y., Bugmann, G., Easterbrook, D.J., 2001. Neural network design for engineering applications. Computers & Structures 79, 1541–1552.
Reckhouse, W., 2010. Optimisation of Short Term Conflict Alert Safety Related Systems. University of Exeter.
RiverCast, P., 2007. Philly RiverCast [WWW Document]. Philly RiverCast. URL http://www.phillyrivercast.org/ (accessed 9.2.14).
Roberts, S., Everson, R., 2001. Independent Component Analysis: Principles and Practice. Cambridge University Press.
Rochester, N., Holland, J., Haibt, L., Duda, W., 1956. Tests on a cell assembly theory of the action of the brain, using a large digital computer. Information Theory, IRE Transactions on 2, 80–93.
Rodriguez, J.J., Kuncheva, L.I., Alonso, C.J., 2006. Rotation Forest: A New Classifier Ensemble Method. IEEE Transactions on Pattern Analysis and Machine Intelligence 28, 1619–1630. doi:10.1109/TPAMI.2006.211
Rogers, L.L., Dowla, F.U., 1994. Optimization of groundwater remediation using artificial neural networks with parallel solute transport modeling. Water Resources Research 30, 457–481. doi:10.1029/93WR01494
Rosenblatt, F., 1960. Perceptron simulation experiments. Proceedings of the IRE 48, 301–309.
Rosenblatt, F., 1958. The perceptron: a probabilistic model for information storage and organization in the brain. Psychological Review Vol. 65, No. 6 386–408.
Rumelhart, D.E., McClelland, J.L., 1986a. Parallel Distributed Processing - Explorations in the Microstructure of Cognition. MIT Press, Cambridge MA.
Rumelhart, D.E., McClelland, J.L., 1986b. Distributed Representations (Hinton Diagrams), in: Parallel Distributed Processing - Explorations in the Microstructure of Cognition. MIT Press, Cambridge MA, pp. 77–109.
Safavian, S.R., Landgrebe, D., 1991. A survey of decision tree classifier methodology. IEEE transactions on systems, man, and cybernetics 21, 660–674.
Sarle, W.S., 1994. Neural networks and statistical models.
Saul, L.K., Roweis, S.T., 2003. Think globally, fit locally: unsupervised learning of low dimensional manifolds. The Journal of Machine Learning Research 4, 119–155.
Savić, D.A., Bicik, J., Morley, M.S., Duncan, A., Kapelan, Z., Djordjevic, S., Keedwell, E.C., 2013. Intelligent Urban Water Infrastructure Management. JIISc, Water Management in Changing Environment 93, 319–335.
Schapire, R.E., 1990. The strength of weak learnability. Mach Learn 5, 197–227. doi:10.1007/BF00116037
Schellart, A.N.A., Rico-Ramirez, M.A., Liguori, S., Saul, A.J., 2009. Quantitative precipitation forecasting for a small urban area: use of radar nowcasting, in: 8th International Workshop on Precipitation In Urban Areas. -, St Moritz, CH, pp. 22–26.

Schellart, A., Ochoa, S., Simões, N., Wang, L.P., Rico-Ramirez, M., Liguori, S., Duncan, A., Chen, A.S., Keedwell, E., Djordjević, S., others, 2011. Urban pluvial flood modelling with real time rainfall information–UK case studies, in: ICUD 2011. Presented at the 12nd International Conference on Urban Drainage, IWA, Porto Alegre/Brazil.

Shen, Q., Diao, R., Su, P., 2012. Feature selection ensemble, in: Proceedings of the Alan Turing Centenary Conference.

Shere, M., 2012. Investigating The Effect Of Data Complexity On Artificial Neural Network Architecture And Performance.

Sindhwani, V., Rakshit, S., Deodhare, D., Erdogmus, D., Principe, J.C., Niyogi, P., 2004. Feature selection in MLPs and SVMs based on maximum output information. Neural Networks, IEEE Transactions on 15, 937–948.

Sjöberg, J., Zhang, Q., Ljung, L., Benveniste, A., Delyon, B., Glorennec, P.-Y., Hjalmarsson, H., Juditsky, A., 1995. Nonlinear black-box modeling in system identification: a unified overview. Automatica 31, 1691–1724. doi:10.1016/0005-1098(95)00120-8

Smith, P., Beven, K.J., Tawn, J.A., 2008. Informal likelihood measures in model assessment: Theoretic development and investigation. Advances in Water Resources 31, 1087–1100. doi:10.1016/j.advwatres.2008.04.012

Solomatine, D., 2008. Data-driven modelling: some past experiences and new approaches. Journal of Hydroinformatics 10.1 3–22.

Solomatine, D., 2007a. Baseflow Separation Techniques for modular ANN modelling in flow forecasting. Hydrological Sciences 52(3) 491–507.

Solomatine, D., 2007b. Knowledge-based modularization and global optimization of artificial neural network models in hydrological forecasting. Neural Networks, Special issue 1–9.

Sontag, E.., 1991. Remarks on interpolation and recognition using neural nets, in: Advances in Neural Information Processing Systems 3 (R.P. Lippmann, J. Moody, and D.S. Touretzky, Eds),. Morgan Kaufmann, San Mateo, CA, pp. 939–945.

Sontag, E.D., 1993. Neural Networks for Control, in: Trentelman, H.L., Willems, J.C. (Eds.), Essays on Control. Birkhäuser Boston, Boston, MA, pp. 339–380.

South West Water, 2014. Beachlive Bathing Water Quality Information System [WWW Document]. Beachlive Bathing Water Quality Information System. URL http://www.beachlive.co.uk/ (accessed 6.23.14).

Specht, D.F., 2006. GRNN with Double Clustering, in: International Joint Conference on Neural Networks, 2006. IJCNN '06. Presented at the International Joint Conference on Neural Networks, 2006. IJCNN '06, pp. 5074–5079. doi:10.1109/IJCNN.2006.247235

Specht, D.F., 1991. A general regression neural network. Neural Networks, IEEE Transactions on 2, 568–576.

Stidson, R.T., Gray, C.A., McPhail, C.D., 2012. Development and use of modelling techniques for real-time bathing water quality predictions. Water and Environment Journal 15.

Sudheer, K.P., Jain, A., 2004. Explaining the internal behaviour of artificial neural network river flow models. Hydrological Processes 18, 833–844. doi:10.1002/hyp.5517

Suñer, D., Malgrat, P., Gutiérrez, E., Clochard, B., 2007. COWAMA (Coastal Water Management) integrated and real time management system of urban water cycle to protect the quality of bathing waters, in: 6th International Conference on Sustainable Techniques and Strategies in Urban Water Management (NOVATECH 2007), Lyon, France.

Thordarson, F.Ø., Madsen, H., 2012. Grey Box Modelling of Hydrological Systems: With Focus on Uncertainties. Technical University of DenmarkDanmarks Tekniske Universitet, AdministrationAdministration, Office for Study Programmes and Student AffairsAfdelingen for Uddannelse og Studerende.

Thorndahl, S., Rasmussen, M.R., Grum, M., Neve, S.L., 2009. Radar Based Flow and Water Level Forecasting in Sewer Systems: a danisk case study.

Tiwari, M., Chatterjee, C., 2010a. Uncertainty assessment and ensemble flood forecasting using bootstrap based artificial neural networks (BANNs). Journal of Hydrology 382, 20–33. doi:10.1016/j.jhydrol.2009.12.013

Tiwari, M., Chatterjee, C., 2010b. Development of an accurate and reliable hourly flood forecasting model using wavelet–bootstrap–ANN (WBANN) hybrid approach. Journal of Hydrology 394, 458–470.

Toth, Z., Kalnay, E., 1993. Ensemble Forecasting at NMC: The Generation of Perturbations. Bulletin of the American Meteorological Society 74, 2317–2330. doi:10.1175/1520-0477(1993)074<2317:EFANTG>2.0.CO;2

Twigt, D., Rego, J.L., Tyrrell, D., Troost, T., 2011. Water Quality Forecasting Systems: Advanced Warning of Harmful Events and Dissemination of Public Alerts, in: Proceedings of the 8th International ISCRAM Conference–Lisbon.

Tyrrell, D., 2010. Bathing Water Quality Forecasting System 2009 Trial, Modelling Report (Internal, unpublished). Environment Agency.

UK Hydrographic Office, 2014. Admiralty Nautical Paper Publications [WWW Document]. The United Kingdom Hydrographic Office. URL https://www.ukho.gov.uk/ProductsandServices/PaperPublications/Pages/NauticalPubs.aspx (accessed 6.25.14).

UKWIR, 2012. The Use of Artificial Neural Networks (ANNs) in Modelling Sewerage Systems for Management in Real Time: Volume 1 - Main Report (12/SW/01/2) (Project Final Report No. 12/SW/01/2). UKWIR (UK Water Industry Research), London, UK.

Ultsch, A., 1993. Self-Organizing Neural Networks for Visualisation and Classification, in: Opitz, P.D.O., Lausen, D.B., Klar, P.D.R. (Eds.), Information and Classification, Studies in Classification, Data Analysis and Knowledge Organization. Springer Berlin Heidelberg, pp. 307–313.

Van Wagner, C.E., 1974. Structure of the Canadian Forest Fires Weather Index. Canadian Forestry, Petawawa Forest Experiment Station, Chalk River, Ontario, Canada.

Verworn, H.R., Krämer, S., 2005. Aspects and Effectiveness of Real-Time Control in Urban Drainage Systems combining Radar Rainfall Forecasts, Linear Optimization and Hydrodynamic Modelling, in: Proc. 8th Int. Conf. on Computing and Control for the Water Industry. pp. 5–7.

Wang, B.X., Japkowicz, N., 2008. Boosting support vector machines for imbalanced data sets, in: Foundations of Intelligent Systems. Springer, pp. 38–47.

Wang, P., Smeaton, A., Lao, S., O'Connor, E., Ling, Y., O'Connor, N., 2009. Short-Term Rainfall Nowcasting: Using Rainfall Radar Imaging. Eurographics ireland pp.

Wang, S., Chen, H., Yao, X., 2010. Negative correlation learning for classification ensembles, in: Neural Networks (IJCNN), The 2010 International Joint Conference on. pp. 1–8.

Wang, X., Yang, J., Teng, X., Xia, W., Jensen, R., 2007. Feature selection based on rough sets and particle swarm optimization. Pattern Recognition Letters 28, 459–471.

Watson, S.W., Novitsky, T.J., Quinby, H.L., Valois, F.W., 1977. Determination of bacterial number and biomass in the marine environment. Applied and Environmental Microbiology 33, 940–946.

White, H., 1989. Learning in Artificial Neural Networks: A Statistical Perspective. Neural Computation 1, 425–464. doi:10.1162/neco.1989.1.4.425

Whitley, D., 2001. An overview of evolutionary algorithms: practical issues and common pitfalls. Information and Software Technology 43, 817–831. doi:10.1016/S0950-5849(01)00188-4

Whitley, D., Starkweather, T., Bogart, C., 1990. Genetic algorithms and neural networks: optimizing connections and connectivity. Parallel Computing 14, 347–361. doi:10.1016/0167-8191(90)90086-O

Whitley, L.D., 1989. The GENITOR Algorithm and Selection Pressure: Why Rank-Based Allocation of Reproductive Trials is Best., in: ICGA. pp. 116–123.

Wikimedia, 2015. Neuron. Wikipedia, the free encyclopedia.

Wikimedia Inc, 2015. Conjugate gradient method. Wikipedia, the free encyclopedia.

Wilcox, R.R., 2012. Comparing Multiple Dependent Groups, in: Introduction to Robust Estimation and Hypothesis Testing. Academic Press, p. 442.

Wilson, E.M., 1990. Engineering hydrology. Macmillan Indianapolis, Indiana,, USA.

Witten, I.H., Frank, E., 2005. Data Mining: Practical machine learning tools and techniques. Morgan Kaufmann.

Wold, S., Esbensen, K., Geladi, P., 1987. Principal component analysis. Chemometrics and Intelligent Laboratory Systems 2, 37–52. doi:10.1016/0169-7439(87)80084-9

Wroblewski, J., 1995. Finding minimal reducts using genetic algorithms, in: Proceedings of Second International Joint Conference on Information Science. pp. 186–189.

Wyer, M.D., O'Neill, G., Ka, D., Crowther, J., Jackson, G., Fewtrell, L., 1997. Non-outfall sources of faecal indicator organisms affecting the compliance of coastal waters with directive 76/160/EEC. Water Science and Technology, Health-Related Water Microbiology 1996 Selected Proceedings of the IAWQ 8th International Symposium on Health-related Water Microbiology 1996 35, 151–156. doi:10.1016/S0273-1223(97)00251-5

Yang, H.H., Amari, S., 1997. Adaptive Online Learning Algorithms for Blind Separation: Maximum Entropy and Minimum Mutual Information. Neural Computation 9, 1457–1482. doi:10.1162/neco.1997.9.7.1457

Yang, J.-B., Shen, K.-Q., Ong, C.-J., Li, X.-P., 2009. Feature Selection for MLP Neural Network: The Use of Random Permutation of Probabilistic Outputs. IEEE Transactions on Neural Networks 20, 1911–1922. doi:10.1109/TNN.2009.2032543

Yang, Q., Shao, J., Scholz, M., Plant, C., 2011. Feature selection methods for characterizing and classifying adaptive Sustainable Flood Retention Basins. Water Research 45, 993–1004. doi:10.1016/j.watres.2010.10.006

Yao, X., 1999. Evolving artificial neural networks. Proceedings of the IEEE 87, 1423–1447.

Yao, X., 1993. A review of evolutionary artificial neural networks. International Journal of Intelligent Systems 8, 539–567. doi:10.1002/int.4550080406

Yu, C.-C., Liu, B.-D., 2002. A backpropagation algorithm with adaptive learning rate and momentum coefficient, in: Proceedings of the 2002 International Joint Conference on Neural Networks, 2002. IJCNN '02. Presented at the Proceedings of the 2002 International Joint Conference on Neural Networks, 2002. IJCNN '02, pp. 1218–1223. doi:10.1109/IJCNN.2002.1007668

Yu, L., Wang, S., Lai, K.K., 2008. Credit risk assessment with a multistage neural network ensemble learning approach. Expert Systems with Applications 34, 1434–1444. doi:10.1016/j.eswa.2007.01.009

Yu, X.-H., Chen, G.-A., 1997. Efficient Backpropagation Learning Using Optimal Learning Rate and Momentum. Neural Networks 10, 517–527. doi:10.1016/S0893-6080(96)00102-5

Zhang, B.-T., Kim, J.-J., 2000. Comparison of selection methods for evolutionary optimization. Evolutionary Optimization 2, 55–70.

Zhang, Q., Stanley, S., 1999. Real-Time Water Treatment Process Control with Artificial Neural Networks. Journal of Environmental Engineering 125, 153–160. doi:10.1061/(ASCE)0733-9372(1999)125:2(153)

Zhang, Z., Deng, Z., Rusch, K.A., 2012. Development of predictive models for determining enterococci levels at Gulf Coast beaches. Water Research 46, 465–474. doi:10.1016/j.watres.2011.11.027

Zhou, A., Qu, B.-Y., Li, H., Zhao, S.-Z., Suganthan, P.N., Zhang, Q., 2011. Multiobjective evolutionary algorithms: A survey of the state of the art. Swarm and Evolutionary Computation 1, 32–49. doi:10.1016/j.swevo.2011.03.001

Zhou, Z.-H., Li, N., 2010. Multi-information Ensemble Diversity, in: Gayar, N.E., Kittler, J., Roli, F. (Eds.), Multiple Classifier Systems, Lecture Notes in Computer Science. Springer Berlin Heidelberg, pp. 134–144.

Zhu, Z., Ong, Y.-S., Dash, M., 2007. Wrapper–filter feature selection algorithm using a memetic framework. Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on 37, 70–76.

Zitzler, E., Deb, K., Thiele, L., 2000. Comparison of multiobjective evolutionary algorithms: Empirical results. Evolutionary computation 8, 173–195.

Zoppou, C., 2001. Review of urban storm water models. Environmental Modelling & Software 16, 195–231. doi:10.1016/S1364-8152(00)00084-0

Appendix A

Bathing beach profiles

Profiles for the 5 designated beaches included in the Chapter 5 Bacti trials follow: Each beach has an online profile provided by the Environment Agency for which a link is provided.

Maps are produced for academic purposes only using Edina DigiMap:

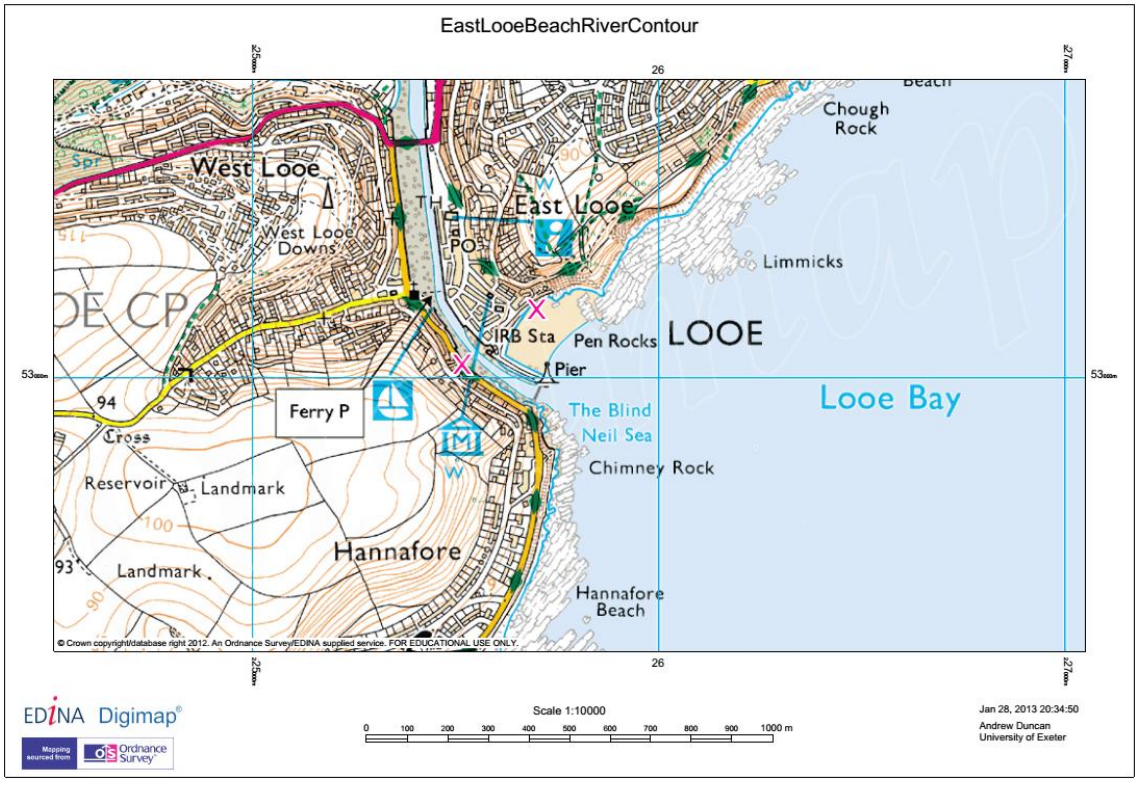
East Looe

(Environment Agency, 2015a)

Beach sample point

River sample point

Map (sample points marked X)

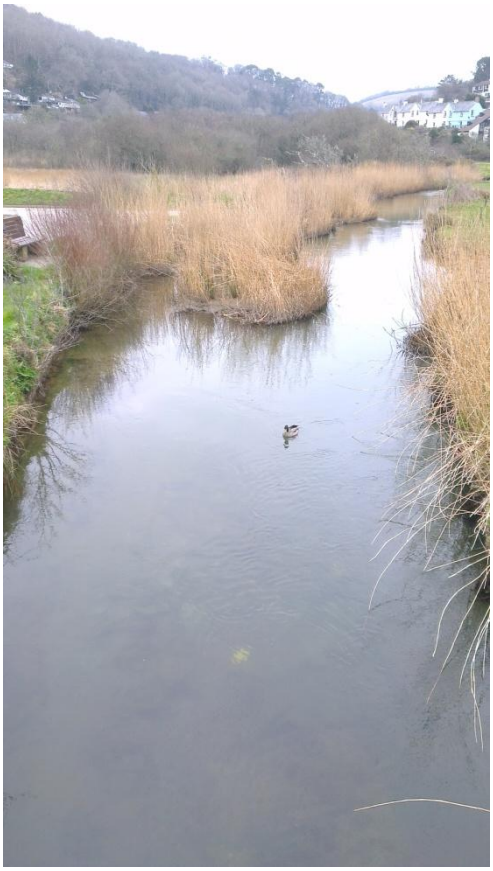


Seaton (Cornwall)

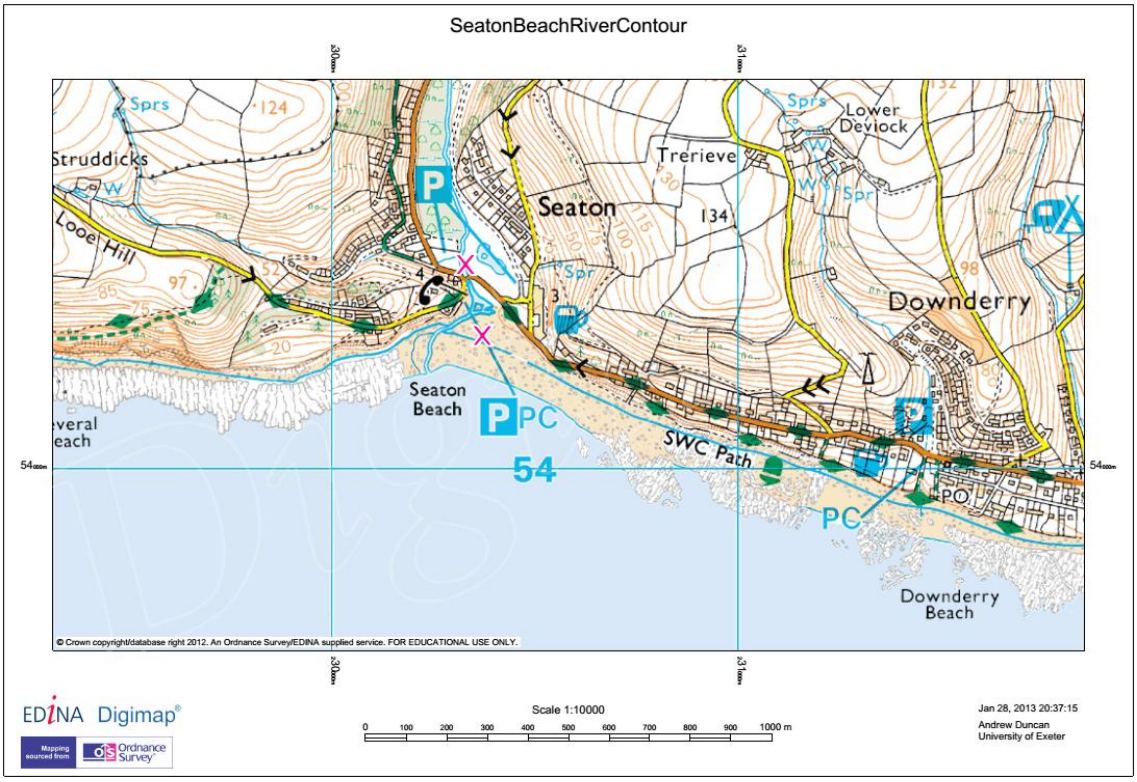
(Environment Agency, 2015b)

Beach sample point

River sample point



Map (sample points marked X)



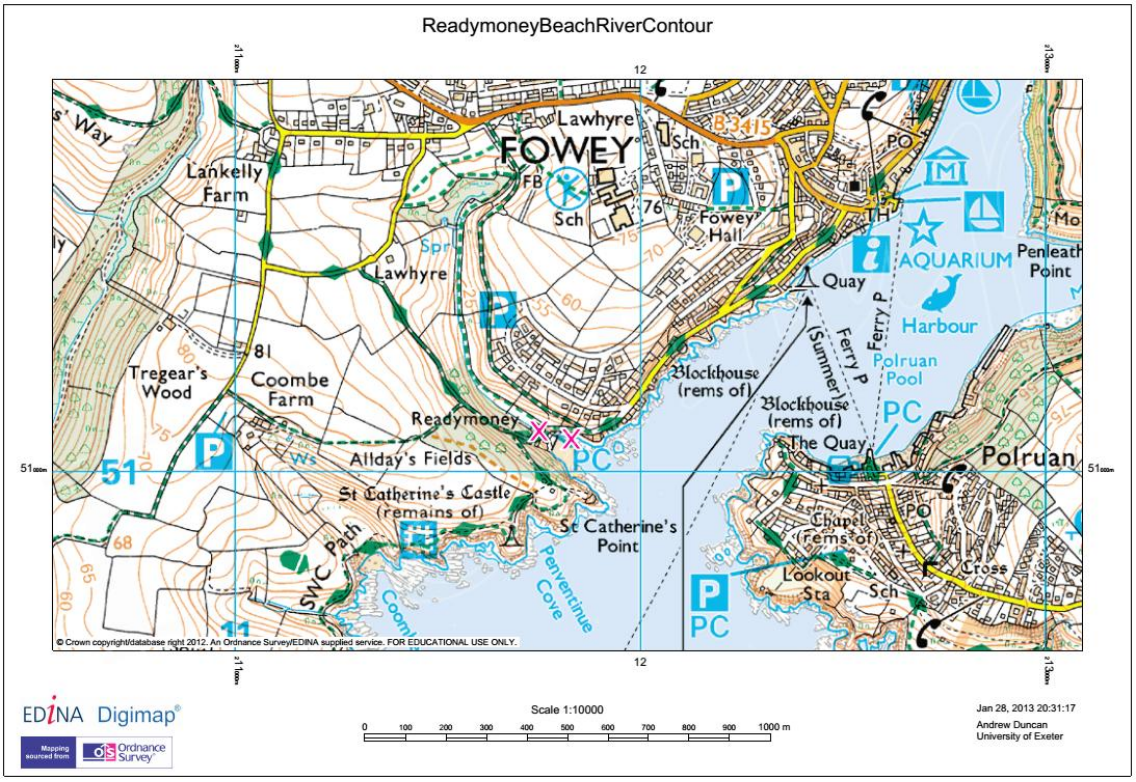
Readymoney

(Environment Agency, 2015c)

Beach sample point

River sample point

Map (sample points marked X)



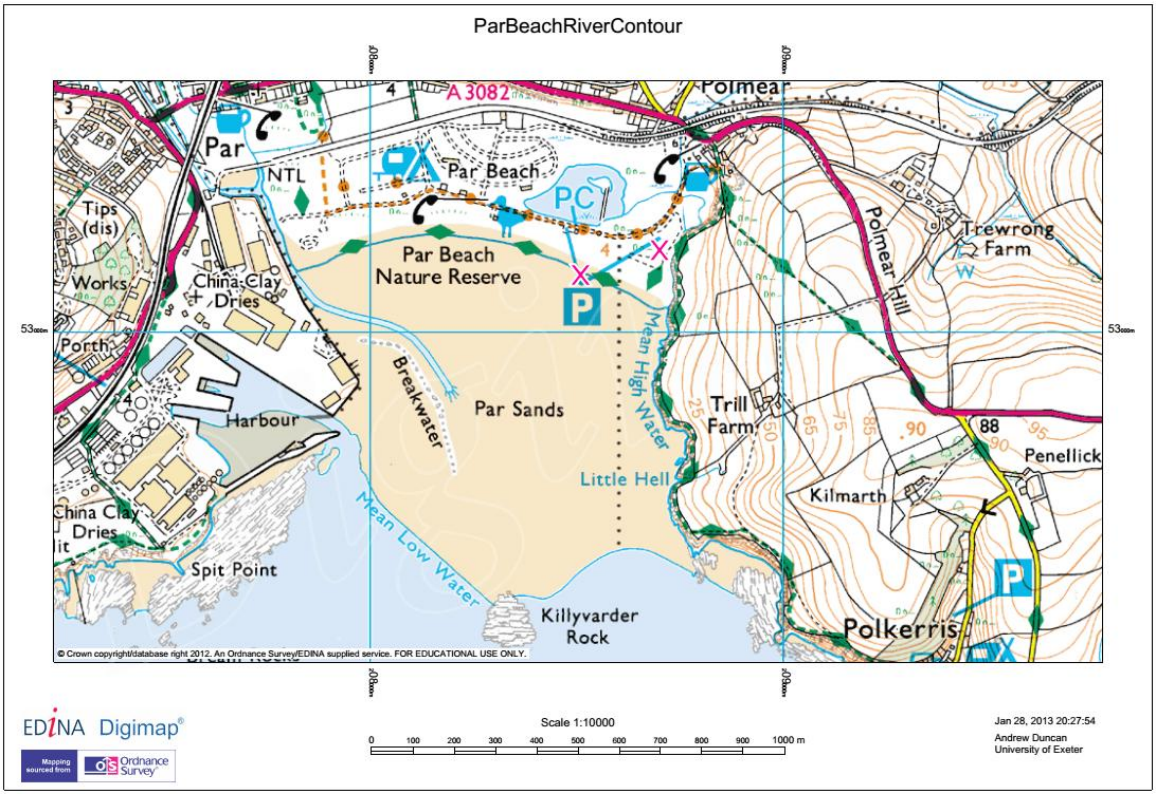
Par

(Environment Agency, 2015d)

Beach sample point

River sample point

Map (sample points marked X)



Porthluney

(Environment Agency, 2015e)

Beach sample point

River sample point

Map (sample points marked X)